On-Line Routing

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Lipmann, Maarten

On-Line Routing / by Maarten Lipmann. - Eindhoven : Technische Universiteit Eindhoven, 2003.
Proefschrift. - ISBN 90-386-0712-1
NUR 918
Subject headings : combinatorial optimisation / routing / travelling salesman problem / optimisation ; algorithms
2000 Mathematics Subject Classification : 90B06, 90C27

Printed by Universiteitsdrukkerij Technische Universiteit Eindhoven

On-Line Routing

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
Rector Magnificus, prof.dr. R.A. van Santen, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op donderdag 19 juni 2003 om 14.00 uur

door

Maarten Lipmann

geboren te Lisse

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr. J.K. Lenstra
en
prof.dr. G.J. Woeginger

Copromotor:
dr. L. Stougie

Preface

About two years ago I started as a Ph.D. student at Eindhoven University of Technology under the supervision of Leen Stougie and Jan Karel Lenstra. My research was on on-line routing. When I started I already had some results that I could use for my thesis, therefore the planning was to finish my Ph.D. in two years. The fact that today my thesis is finished proves that this planning was right.

This thesis would probably not exist if Leen Stougie had not taken the effort to convince me to become a Ph.D. student. We met at the University of Amsterdam, where he supervised my master thesis. By inspiring and challenging me Leen made me think harder and in a more structured way than I used to do. I thank Leen for his enthusiasm and interest in my work and for the good times we had outside working hours.

I also want to thank Jan Karel Lenstra. In the first year he told me to write down my results as soon as possible. I knew he was right, but I had other 'priorities'. I wish I had listened better. I learned a lot about writing through his comments on early versions of papers and of this thesis. I admire his ability to explain complex things in a transparent and clear manner.

I am also grateful to my colleagues Willem de Paepe, Rene Sitters, and Xiwen Lu. We have spent quite some time together, not only in Eindhoven, but also in Rome, Lunteren, and Como during workshops and conferences. It was a real pleasure working with them.

Finally, I want to thank my family and friends for their support and interest in my work.

Maarten Lipmann
Amsterdam, April 2003.

Contents

1

Introduction

This thesis is about routing problems in which the requests are presented over time.

Routing problems involve the design of routes to serve a sequence of requests for rides or visits to points in some metric space, in order to achieve optimality. Historically, routing problems are studied from the off-line point of view; the entire input sequence is given beforehand. However, in practice often information about the requests becomes available over time, in an on-line fashion. Think for example of an elevator: over time it receives requests for transportation of people from one floor to another. This on-line model also accommodates on-line routing problems that occur in practice for taxis and courier services.

We consider two types of on-line routing problems. The first one is the on-line travelling salesman problem, in which each request is a point requested to be visited. The second one is the on-line dial-a-ride problem, in which each request is a ride that has to be executed. We note that the on-line travelling salesman problem is a special case of the on-line dial-a-ride problem in which all rides have length zero.

This introductory chapter is organized as follows. In Section 1.1 we explain the difference between on-line and off-line. Next, in Section 1.2 we explain and discuss competitive analysis, which we use to measure the quality of on-line algorithms. In Section 1.3 we introduce the on-line travelling salesman problem and in Section 1.4 the on-line dial-a-ride problem. We conclude this chapter in Section 1.5 with an outline of this thesis.

1.1 On-line versus off-line

Routing problems have been widely studied for more than three decades. The common approach to the problem is the off-line point of view in which all information about the requests is known in advance. However, in practice very often the input sequence is communicated in successive steps and becomes known in an on-line fashion.

There are several models which describe the manner in which the input sequence is given; see for example [7]. The two most common models are the *making decisions one-by-one* model and the *real time* model. In the making decisions one-by-one model the requests are ordered in a list and are presented one by one to the on-line algorithm. The requests must be served in the order of their occurrence. Only after the on-line algorithm serves a request, the next request on the list becomes available. The on-line algorithm does not know the next request on the list, not even if there is a next request. Actions made by an on-line algorithm cannot be revoked.

In the real time model, requests are presented over time. The on-line algorithm has to decide on a tour to visit the requests presented up to that specific time. At any time, new requests may arrive. Thus, at any time the on-line algorithm knows only the requests presented in the past and does not know any future request, not even if there will be any. Time flows while decisions are made and executed; decisions made by an on-line algorithm can only be revoked as long as it did not execute them yet. The on-line model we use throughout this thesis is the real time model.

The real time model is very natural in on-line routing and on-line scheduling, since it allows waiting, postponing decisions, and serving the requests in an order different from the order in which they are presented.

In the first four chapters of this thesis we assume that a request is completely specified the moment it is presented. In Chapter 5 we deviate from this setting by only giving partial information about the request the moment it is presented. The moment the rest of the information about the request becomes known, depends on the behaviour of the on-line algorithm.

1.2 Competitive analysis

For most on-line optimization problems there are no algorithms that attain the optimal off-line objective function value on every input sequence, irrespective of the computation time that is allowed. In theoretical studies *competitive analysis* is the most widely accepted way of measuring the performance of on-line algorithms. It was proposed for the first time in [22]; for a survey see [5]. The worst-case ratio over all possible sequences of requests, between the objective function value produced by an on-line algorithm and that of an optimal off-line algorithm on the same input sequence, is called the *competitive ratio* of the on-line algorithm. Correspondingly, we call an algorithm ρ-competitive if for every input sequence, the objective function value of this algorithm is at most ρ times the objective function value of an optimal algorithm on the same input.

Competitive analysis often employs the notion of an adversary in a two-person game setting (see e.g. [5]). In this setting an adversary plays against an on-line player. We use the real time model; therefore, time is essential for this game setting. The adversary provides an input sequence over time that both players have to process. The game begins at time 0 and the adversary can start presenting requests. At any time the on-line player only knows requests presented in the past and has to process the requests while time is running and new requests may arrive. He can choose any on-line algorithm A from the set $\mathcal{A}$ of on-line algorithms to process the sequence of requests. We restrict ourselves to on-line players that use a deterministic algorithm. The adversary can study the behaviour of the on-line player to decide on the next request, or to decide he presents no more request. After the adversary presented the last request, he has to process the same input sequence. At that point the adversary knows the entire input sequence and can use an optimal off-line algorithm to process this sequence of requests.

The on-line player tries to choose an algorithm $A \in \mathcal{A}$ that minimizes the ratio between his own objective function value $Z^{OL}(A, \sigma)$ and the optimal off-line objective function value $Z^*(\sigma)$ over all possible input sequences σ:

$$\min_{A \in \mathcal{A}} \max_{\sigma \in I} \frac{Z^{OL}(A, \sigma)}{Z^*(\sigma)}.$$

The adversary cannot provide an input sequence to the on-line player who uses the algorithm A for which this ratio is attained, that gives a higher ratio. Hence, this algorithm A gives the best possible competitive ratio for the problem under study. We call such an on-line algorithm *best possible*.

In order to decide if an on-line algorithm is best possible, a lower bound is derived on the competitive ratio of any on-line algorithm. For this purpose the adversary studies the behaviour of the on-line player over time and, based on this behaviour, tries to present the most inconvenient request sequence for the on-line player. From the set I of possible input sequences, the adversary tries to provide the sequence of requests σ_A to the on-line player who uses algorithm $A \in \mathcal{A}$, that maximizes the ratio between the objective function value of the on-line player $Z^{OL}(A, \sigma_A)$, and his own optimal off-line objective function value $Z^*(\sigma_A)$ on the same input sequence σ_A. The strategy to achieve lower bounds on the competitive ratio of a best possible on-line algorithm is to devise a bad input sequence for every algorithm or a set of such sequences for various classes of algorithms together covering the class of all deterministic algorithms. In this way the highest possible lower bound will be equal to the competitive ratio of a best possible on-line algorithm.

In competitive analysis every on-line algorithm is judged in the same way, but it is judged on a specific aspect only. We ought to be careful if we compare on-line algorithms for the same problem, based on their competitive ratios only. To say that a certain on-line algorithm performs better because it has a better competitive ratio, may be viewed as not realistic.

First of all, the competitive ratio is by definition a worst-case ratio. An algorithm that works well on almost all possible input sequences is judged only on its bad performance on a typical input sequence. The emphasis on a worst-case instance can be seen as being too pessimistic; an average case ratio may be more realistic. But, then again, in average case analysis we use a probability distribution for the input sequences, which makes average case analysis no more realistic than the probability distribution we choose. Second, the emphasis on the worst-case instance brings along the risk of tailoring algorithms so as not to fail on some particular instances, which may lead to unrealistic behaviour of the on-line algorithm on instances that are more typical for real life instances. Third, we do not bother about complexity issues and allow unrestricted running time of an algorithm. This might make sense from a theoretical point of view; for practical applications it is not realistic.

Another reason why the comparison of on-line algorithms for the same problem, based on their competitive ratios only, may be viewed as not realistic, is the (arguably) unrealistic power of the adversary against whom the performance of the on-line algorithms is measured.

A natural approach to get a more realistic performance measure is to restrict the power of the adversary, which at the same time may rule out some unrealistic input sequences. In *comparative analysis*, introduced in [12], the adversary or the on-line player is restricted in his choice of algorithms. We give some examples. Blom et al. [4] introduce a *fair adversary*, who is restricted in his movements. The authors show that fair adversaries are weaker and therefore competitiveness under fairness can be better than if fairness is not imposed. In the same article Blom et al. introduce a particular class of algorithms for on-line routing problems which they call *zealous algorithms*. Roughly speaking, the on-line server should never sit still, as long as there are unserved requests. The restriction on the on-line player to a specific class of algorithms leads to higher competitive ratios.

Krumke et al. [16] introduce a *non-abusive adversary*. The authors show that for the problem they studied, a non-abusive adversary allows for a constant competitive ratio, while against an unrestricted adversary no constant competitive ratio is possible. To restrict the class of on-line algorithms to those having polynomial running time also fits in the comparative analysis framework.

In this thesis, we use competitive analysis as the standard way of measuring the performance of on-line algorithms. We show that some of the extensions of competitive analysis described above are useful for specific problems and powerful enough to get significant results. We also show that combining these approaches can give more insight in the structure of a specific problem and the cost of not having information about future requests.

1.3 The on-line travelling salesman problem

The travelling salesman problem is one of the most extensively studied problems in combinatorial optimization. Given a set of points in some metric space we wish to find a shortest tour visiting all the points and returning to the departure point. A comprehensive survey of the numerous facets of this problem is found in the book edited by Lawler, Lenstra, Rinnooy Kan, and Shmoys [15], or for a more recent work, the book edited by Gutin and Punnen [10]. The problem is NP-hard [11], [8] in general metric spaces, even in $\mathbb{R}^2$, or on a grid graph. It is easy on a tree, following any depth first search, and trivial to solve if the metric space is the real line: going first to the leftmost extreme, then to the rightmost extreme, and finally back to the origin.

The problem becomes more complicated if we consider the situation of the salesman wherein he does not have all information in advance. If the points to be visited are not known in advance but revealed while the salesman has started his tour already, the problem becomes the *on-line travelling salesman problem* (OLTSP). The server has to make a tour to visit these points. While the server is on his tour, new requests may arrive. Thus, at any time the server knows only the points requested in the past and does not know any future request, not even if there will be any future requests. The objective is to serve all requests as soon as possible, i.e., to minimize the completion time.

Ausiello et al. [3] posed and studied the OLTSP. They called the problem in which the server is to return to the departure point after having visited all requested points the *Homing*-OLTSP (HOLTSP), as opposed to the *Nomadic*-OLTSP (NOLTSP), in which the endpoint of his tour is free. Notice that the off-line version of this problem is actually a travelling salesman problem with individual release times of the points, i.e., specific moments in time at or after which the salesman must visit the points. This off-line problem is of course also NP-hard for general metric spaces, but its complexity is unknown for trees (see [18]). It remains easy for the real line, but much less trivial than the TSP without release times [23].

The *on-line travelling repairman problem* (OLTRP) is a variant of the on-line travelling salesman problem. The objective is to minimize the sum of completion times of the requests. The completion time of a request is defined as the time at which the request is served. The sum of completion times of the requests is also referred to as the *latency*. On-line travelling repairman problems have been studied in [6] and [14]. The off-line version of this problem is a travelling repairman problem with individual release times of the points. The computational complexity of this problem is still unknown [19]. The off-line travelling repairman problem is known to be NP-hard [1]. For the real line the TRP can be solved in polynomial time [1]. Recently, Sitters [21] showed that the TRP on trees is NP-hard.

1.4 The on-line dial-a-ride problem

In on-line dial-a-ride problems, servers are travelling in some metric space to serve requests for rides that are presented over time. Each ride is characterized by two points in the metric space, a *source*, the starting point of the ride, and a *destination*, the end point of the ride. As in the on-line travelling salesman problem, the objective is to find a tour for the servers that finishes as early as possible. On-line dial-a-ride problems have been studied in [2] and [6].

Each ride stands for the transportation of an item. The capacity of a server is defined as the number of rides (items) it can execute (transport) simultaneously. If preemption is allowed, a server is allowed to preempt any ride at any point and resume the ride in that point later. We note that for the special case in which all rides have length zero (i.e., an instance of the travelling salesman problem) the capacity and preemption are irrelevant. There is a great variety of dial-a-ride problems. In [20] de Paepe et al. examined the computational complexity of dial-a-ride problems and proposed a classification similar to the one developed for scheduling problems [9].

As for the OLTSP, we distinguish between the Homing-OLDARP (HOLDARP), in which the server is to return to the origin after having executed all rides, and the Nomadic-OLDARP (NOLDARP), in which the endpoint of his tour is free.

The *latency on-line dial-a-ride problem* (L-OLDARP) is an on-line dial-a-ride problem with the objective to minimize the sum of completion times of the rides. The completion time of a ride is defined as the time at which the ride is finished. Latency on-line dial-a-ride problems have been studied in [6] and [14]. The on-line travelling repairman problem is a latency on-line dial-a-ride problem in which all rides have length zero.

In [2] and [6] on-line dial-a-ride problems are studied in which information about a ride (i.e., source and destination) becomes available the moment it is presented. In *the restricted information model* (see Chapter 5) we deviate from this setting by only revealing the source of a ride the moment it is presented; the destination is only revealed the moment the ride is picked up in the source.

1.5 Outline of the thesis

In Chapter 2, we define the on-line travelling salesman problem and the on-line dial-a-ride problem more formally and introduce notation. We specify the class of problems that we investigate. We give an overview of all results on the research on the on-line travelling salesman problem and the on-line dial-a-ride problem and indicate which results are presented in this thesis.

In Chapter 3, we present lower bounds on the competitive ratio for the on-line travelling salesman problem and the on-line dial-a-ride problem. In Chapter 4, we present algorithms with the currently best known competitive ratio for the on-line travelling salesman problem and the on-line dial-a-ride problem. Among others we give best possible algorithms for the HOLTSP on the real line. Parts of Chapter 3 and 4 are joint work with W.E. de Paepe and L. Stougie.

Chapter 5, which is based on joint work with X. Lu, W.E. de Paepe, R.A. Sitters, and L. Stougie [17], is about on-line dial-a-ride problems under the restricted information model (RIM). Since the RIM is a specific and unconventional model, we choose to treat the RIM separately. We believe RIM describes some practical situations better than the conventional model.

2

On-line routing

In this chapter we shall be specific about the problems we investigate in this thesis. In Section 2.1 we introduce notation and give more formal definitions of the problems considered in this thesis. Next, in Section 2.2 we specify the type of algorithms studied in this thesis and the adversaries against whom we measure the performance of the algorithms. In Section 2.3 we give the state of the art on research on the on-line routing problems considered. We conclude in Section 2.4 with a discussion of open problems and possible solutions.

2.1 Notation and definitions

In this section we introduce notation and define the problems under consideration.

Metric space. Throughout this thesis, we use the metric space $M = (X, d)$ with a special point $O \in X$ selected as the origin. We assume that M has the property that for any pair of points $\{x, y\} \in X$ there is a continuous path $p\colon [0, 1] \to X$ with $p(0) = x$ and $p(1) = y$ of length $d(x, y)$ (see [3] for a thorough discussion of this model). We assume that d is symmetric and satisfies the triangle inequality. We distinguish between general metric spaces, the real line, and the halfline. If the metric space is the real line or the halfline, we consider the origin O at point 0.

We define the problems we investigate.

On-line travelling salesman problem. We define the on-line single server travelling salesman problem (OLTSP) as the problem of one server travelling in some metric space M. The server is in the origin at time 0. The server can travel at maximum at unit speed. Over time requests are presented. Each request is a pair $\sigma_i = (t_i, x_i)$, where $t_i \in \mathbb{R}_0^+$ is the time at which request σ_i is released, and $x_i \in X$ is the point in the metric space requested to be visited. We assume that the sequence $\sigma = \sigma_1, \ldots, \sigma_m$ of requests is revealed in order of non-decreasing release times, and that the on-line server has neither information about the time when the last request is released, nor about the total number of requests. For $t \geq 0$ we denote by $\sigma_{\leq t}$ the set of requests in σ released no later than time t. An on-line algorithm for the OLTSP must determine the behavior of the server at any moment t as a function of t and $\sigma_{\leq t}$. A feasible solution is a route for the server that starts in the origin O and serves all requests such that each request is served not earlier than the time it is released. The objective is to find a tour for the server that finishes as early as possible i.e., to minimize the completion time.

The problem in which the server is to return to the origin after having visited all requested points is called the *Homing*-OLTSP (HOLTSP), as opposed to the *Nomadic*-OLTSP (NOLTSP), in which the endpoint of his tour is free.

On-line dial-a-ride problem. We define the on-line single server dial-a-ride problem (OLDARP) as the problem of one server travelling in some metric space M with a special point O selected as the origin. The server is in the origin at time 0. He can travel at maximum at unit speed. Over time requests for rides are presented. Each *ride* is a triple $\sigma_i = (t_i, s_i, d_i)$, where $t_i \in \mathbb{R}_0^+$ is the time at which ride σ_i is released, $s_i \in X$ is the source of the ride, and $d_i \in X$ is the destination of the ride. Every ride $\sigma_i \in \sigma$ has to be executed (served) by the server, that is, he has to visit the source, start the ride, and end it at the destination. The *capacity* of the server is an upper bound on the number of rides he can execute simultaneously. We only consider unit capacity for the server. We do not allow preemption, so once the server picks up a ride in its source, he must execute it completely before he can start executing other rides. As for the OLTSP, the objective is to find a tour for the server that finishes as early as possible (to minimize the completion time). We assume that the sequence $\sigma = \sigma_1, \ldots, \sigma_m$ of rides is revealed in order of non-decreasing release times, and that the on-line server has neither information about the time when the last ride is released, nor about the total number of rides. For $t \geq 0$ we denote by $\sigma_{\leq t}$ the set of rides in σ released no later than time t. An on-line algorithm for the OLDARP must determine the behavior of the server at any moment t as a function of t and $\sigma_{\leq t}$, whereas the off-line algorithm knows σ at time 0. A feasible solution is a route for the server that starts in the origin O and serves all requested rides such that each ride is picked up at the source not earlier than the time it is released.

In the *Homing*-OLDARP (HOLDARP) the server is to return to the origin after having executed all requested rides. In the *Nomadic*-OLDARP (NOLDARP), the endpoint of the tour is free.

On-line travelling repairman problem and latency on-line dial-a-ride problem. The on-line travelling repairman problem (OLTRP) is a variant of the on-line travelling salesman problem. Instead of minimizing the overall completion time, the objective is now to minimize the sum of the completion times of the requests. The completion time of a request is defined as the time at which the request is served. This objective is also referred to as the *latency*.

The latency on-line dial-a-ride problem (L-OLDARP) is an on-line dial-a-ride problem with the objective to minimize the sum of completion times of the rides. The completion time of a ride is defined as the time at which the ride is finished.

Since we minimize the sum of the completion times of the requests, we do not distinguish between the problem in which it is required to return to the origin after having served all the requests, and the problem in which the endpoint of the tour is free.

2.2 Algorithms and adversaries

We restrict ourselves to deterministic algorithms in which the on-line player chooses his strategy deterministically. A deterministic algorithm produces the same output and has the same objective function value, every time it is faced with the same input sequence. In randomized algorithms, the on-line player can choose his strategy at random. For instance, he can make a random choice from the set of deterministic algorithms every time he has to make a decision. The output produced by a randomized algorithm is random and the cost incurred a random variable.

The server used by a deterministic algorithm can travel at any speed ranging from 0 to unit speed. He can adjust his speed or change his direction at any time and at any place. More specifically, he is allowed to sit and wait while there are still unserved requests. In [4] Blom et al. introduce a particular class of algorithms for on-line routing problems which they call *zealous algorithms*.

Zealous algorithm. The server used by a zealous algorithm, a zealous server, should never sit and wait when he could serve unserved requests. If there are still unserved requests, then the direction of a zealous server changes only if a new request becomes known, or the server is either in the origin or at a request that has just been served. A zealous server moves only at maximum (i.e. unit) speed and does not make any detours.

We use competitive analysis to measure the performance of on-line algorithms (see Section 1.2). We see competitive analysis as a game between an on-line player and a malicious adversary. The adversary provides the input sequence over time. He can study the behaviour of the on-line player to decide on an inconvenient next request for the on-line player. The on-line player does not know the future and has to process the requests while time is running and new requests may arrive. The adversary has to process the same sequence of requests, but not at the same time he presents it. He can process the input when it is completely known, in an optimal off-line manner. Each request has a release time at which it is presented. The adversary knows the input sequence when he starts processing it, so he can go to the point of the next request before the release time of the request. The unnatural advantage for the adversary of being able to go to requests before their release time combined with the fact that not even a smart on-line algorithm can anticipate those requests, led Blom et al. [4] to introduce a notion of *fairness* in competitive analysis of on-line routing problems.

Fair adversary. At any moment t, the off-line optimal tour, taken by the adversary, is not allowed to move outside the convex hull of the origin O and the requested points from $\sigma_{<t}$.

We investigate the influence of introducing fairness and the influence of restricting ourselves to zealous on-line algorithms on the competitiveness for all problems defined in Section 2.1. Since it is not clear how to define the convex hull on graphs, we do not consider fairness in general metric spaces.

2.3 Overview of best known results

In this section we give the state of the art for the problems considered in this thesis. We refer to the literature or give the number of the theorem if the result is ours. An overview of the results discussed in this section is given in the tables in Section 2.3.6.

Some results for a particular problem imply the same result for other problems. W. E. de Paepe et al. [20] introduced a systematic use of these kinds of implications for dial-a ride problems. We start with the following observations. An on-line travelling salesman problem is a special case of the on-line dial-a-ride problem. Therefore, all the lower bound results for the OLTSP also hold for the OLDARP and the upper bound results for the OLDARP hold for the OLTSP.

The class of general metric spaces contains the real line as a special case, and the real line contains the halfline as a special case. So, any lower bound result for the halfline holds for the real line and general metric spaces; any lower bound result for the real line also holds for general metric spaces. Any upper bound result for general metric spaces holds for the real line and the halfline; any upper bound result for the real line also holds for the halfline.

Any lower bound result under the fairness restriction also holds if fairness is not imposed; any upper bound result without the fairness restriction also holds if fairness is imposed.

Any lower bound result without the zealousness restriction holds for the case with the zealousness restriction; any upper bound result under the zealousness restriction holds for the case without the zealousness restriction.

2.3.1 The HOLTSP

The best known results for the HOLTSP discussed in this section are summarized in Table 2.1.

For the HOLTSP on general metric spaces a lower bound of 2 on the competitive ratio of any on-line algorithm is matched by a 2-competitive zealous on-line algorithm in [3]. In Section 3.1 we give an alternative proof for this lower bound

(Theorem 3.1). In the same paper Ausiello et al. prove a lower bound on the competitive ratio of $(9+\sqrt{17})/8 \approx 1.64$ in case the metric space is the real-line. The authors also present a 7/4-competitive zealous algorithm that is best possible within this restricted class, since in [4] it is shown that zealous algorithms cannot have competitive ratios lower than 7/4. The question remained open if the lower bound of $(9+\sqrt{17})/8$ was too low or if there exists an on-line non-zealous algorithm with competitive ratio better than 7/4. In Section 4.1.1 we answer this question in favor of the latter possibility, by providing a best possible algorithm with competitive ratio $(9+\sqrt{17})/8$ (Theorem 4.3). The algorithm is based on a minute study of the lower bound in [3] for deciding when and how long to wait. This result shows that waiting actually helps to get better competitive ratios.

In Section 3.1 we derive a lower bound of $(5+\sqrt{57})/8 \approx 1.57$ on the competitive ratio of any deterministic algorithm for the HOLTSP under the fairness restriction on the real line (Theorem 3.2). In Section 4.1.2 we adjusted the above mentioned $(9+\sqrt{17})/8$-competitive algorithm such that it has a matching competitive ratio for the problem on the real line under the fairness restriction (Theorem 4.4).

In [19] de Paepe proves a lower bound of 8/5 on the competitive ratio of a zealous deterministic algorithm against a fair adversary for the case the metric space is the real line. The best known zealous algorithm is the already mentioned 7/4-competitive zealous algorithm in [3]. This is the only problem for the HOLTSP for which there are no matching lower bound and upper bound.

In [4] Blom et al. give a lower bound of 3/2 on the competitive ratio of any deterministic algorithm for the case the metric space is a half-line, together with a zealous algorithm with matching competitive ratio. If fairness is imposed the authors found a lower bound of $(1+\sqrt{17})/4 \approx 1.28$, together with a non-zealous algorithm with matching competitive ratio.

For the case in which fairness is imposed and is restricted to zealous on-line algorithms, Blom et al. [4] found a lower bound of 4/3, together with a zealous algorithm with matching competitive ratio. This result again shows that waiting can help to get better competitive ratios.

2.3.2 The NOLTSP

The best known results for the NOLTSP discussed in this section are summarized in Table 2.1.

The competitive ratios for the NOLTSP are higher than those for the HOLTSP. Intuitively, this makes sense since having to return to the origin provides the on-line algorithm with extra information.

For the NOLTSP on general metric spaces Ausiello et al. [3] give a lower bound of 2 on the competitive ratio of any deterministic on-line algorithm, together with a 5/2-competitive zealous algorithm. It was believed that this lower bound of 2 was tight. However, in Section 3.2 we prove a lower bound of $x \approx 2.02976$ which is the solution to $3x^3 + 3x^2 - 15x - 7 = 0$ (Theorem 3.3). This lower bound is achieved on the real line. In Section 4.2.1 we present a $(1+\sqrt{2})$-competitive algorithm for general metric spaces, which uses waiting (Theorem 4.5). For zealous algorithms we prove a lower bound on general metric spaces of $(2\sqrt{21}-3)/3 \approx 2.06$ in Section 3.2 (Theorem 3.4). This lower bound is achieved on the real line as well.

We present a 2.06-competitive algorithm in case the metric space is the real line in Section 4.2.2 (Theorem 4.8). Like the algorithm for the HOLTSP on the real line, this algorithm waits until a specific time implied by the lower bound construction. This algorithm is also the best known in case the metric space is the halfline. The 7/3-competitive zealous algorithm for the real line presented in [3] is the best known zealous algorithm in case the metric space is the real line or the halfline.

In Section 3.2 the following lower bounds are presented. If fairness is imposed we prove a lower bound of $(1+\sqrt{97})/6 \approx 1.81$ in case the metric space is the real line (Theorem 3.5). If fairness is imposed and we restrict ourselves to zealous on-line algorithms, we found a lower bound of $\sqrt{33}/3 \approx 1.91$ (Theorem 3.6).

In case the metric space is the halfline, we prove a lower bound of $x \approx 1.63$ which is the solution to $4x^4 - 6x^3 + 3x^2 - 5x - 2 = 0$ (Theorem 3.7). If we restrict ourselves to zealous on-line algorithms, we found a lower bound of 7/4 (Theorem 3.9). In case fairness is imposed, we prove a lower bound of $x \approx 1.60$ which is the solution to $2x^3 + x^2 - 3x - 6 = 0$ (Theorem 3.8). If fairness is imposed and we restrict ourselves to zealous on-line algorithms, we give a lower bound of $\sqrt{3} \approx 1.73$ (Theorem 3.10).

There are no algorithms known designed specifically for the case in which the metric space is the halfline, or for the case in which fairness is imposed. We note that the set of results of the research on the competitiveness of deterministic algorithms for the NOLTSP is not as complete as the set of results for the HOLTSP.

2.3.3 The HOLDARP

The best known results for the HOLDARP discussed in this section are summarized in Table 2.2.

For the HOLDARP on general metric spaces a lower bound of 2 on the competitive ratio of any deterministic on-line algorithm comes from an instance of the HOLTSP [3]. Krumke [13] gives a 2-competitive non-zealous algorithm, matching this lower bound. This algorithm is also the best known in case the metric space is the real line or the halfline.

We present an alternative lower bound of 2 for the zealous case, which is achieved on the halfline against a fair adversary (Theorem 3.11). The best known zealous algorithm has competitive ratio 5/2 [2],[6] (in these independent efforts the same algorithm was found). This algorithm is also the best known in case the metric space is the real line or the halfline.

In case the metric space is the real line we present a lower bound of 7/4 for any deterministic on-line algorithm (Theorem 3.12). If fairness is imposed we prove a lower bound of $(1+\sqrt{5})/2 \approx 1.62$ which is achieved on the halfline (Theorem 3.13). In case the metric space is the halfline a lower bound of $(2+\sqrt{2})/2 \approx 1.71$ for any deterministic on-line algorithm is implied by the result in [13].

Our lower bound results can be found in Section 3.3.

2.3.4 The NOLDARP

The best known results for the NOLDARP discussed in this section are summarized in Table 2.2. Our lower bound results can be found in Section 3.4.

For the NOLDARP on general metric spaces a lower bound of approximately 2.03 on the competitive ratio of any deterministic on-line algorithm comes from an instance of the NOLTSP on the real line (Section 3.2, Theorem 3.3). We present a $(3+\sqrt{5})/2 \approx 2.62$-competitive non-zealous algorithm in Section 4.3 (Theorem 4.9). This algorithm is also the best known in case the metric space is the real line or the halfline. For the zealous case, we present a lower bound of 5/2, which is achieved on the halfline (Theorem 3.15). The best known zealous algorithm has competitive ratio 3 [13]. This algorithm is also the best known in case the metric space is the real line or the halfline.

If fairness is imposed we prove a lower bound of $(1+\sqrt{22})/3 \approx 1.90$ which is achieved on the halfline (Theorem 3.14). If fairness is imposed and we restrict ourselves to zealous on-line algorithms, we found a lower bound of 2 that is achieved on the halfline as well (Theorem 3.16).

2.3.5 The OLTRP and the L-OLDARP

The best known results discussed in this section are summarized in Table 2.3.

We note that fairness does not make sense if the objective is to minimize the sum of completion times. The fair adversary can easily give two single requests at time 0 to buy himself some space. The completion times of these two requests can be neglected if the total number of request gets large enough.

In [6] Feuerstein and Stougie show that restricting to zealous on-line algorithms leads to nonconstant competitive ratios. In the same paper the authors prove a lower bound of 3 for the L-OLDARP on the real line and a lower bound of $1+\sqrt{2}$ for the OLTRP on the real line. For general metric spaces no better lower bounds are known. In Section 3.5 we present two (trivial) lower bounds in case the metric space is the halfline: a lower bound of 2 for the OLTRP (Theorem 3.17) and a lower bound of $1+\sqrt{2}$ for the L-OLDARP (Theorem 3.18).

Krumke et al. [14] present a $(1+\sqrt{2})^2 \approx 5.83$-competitive algorithm for the L-OLDARP in general metric spaces, which is also the best known for the real line or the halfline. The same algorithm is also the best known for the OLTRP in general metric spaces or the real line. In case the metric space is the halfline, the 9-competitive algorithm for the OLTRP on the real line in [6] is 7/2-competitive (with some minor adjustments, as was pointed out by G. J. Woeginger).

2.3.6 Tables with best known upper and lower bounds

In the tables we refer to the literature or give the number of the theorem if the result is ours. If a result is implied by another result, we use an arrow ($\rightarrow$), pointing in the direction of the original result, and either refer to the literature or give the number of the theorem by which the result is implied.

Table 2.1: Lower bounds and upper bounds on the competitive ratio of deterministic algorithms for on-line travelling salesman problems.

		non-zealous		zealous	
		lower bound	upper bound	lower bound	upper bound
Homing general		2 [3]	2 $\rightarrow$ [3]	2 $\leftarrow$ [3]	2 [3]
Homing real line	standard	$\frac{9+\sqrt{17}}{8}$ [3]	$\frac{9+\sqrt{17}}{8}$ (4.3)	$\frac{7}{4}$ [4]	$\frac{7}{4}$ [3]
	fair	$\frac{5+\sqrt{57}}{8}$ (3.2)	$\frac{5+\sqrt{57}}{8}$ (4.4)	$\frac{8}{5}$ [19]	$\frac{7}{4}$ $\uparrow$ [3]
Homing halfline	standard	$\frac{3}{2}$ [4]	$\frac{3}{2}$ $\rightarrow$ [4]	$\frac{3}{2}$ $\leftarrow$ [4]	$\frac{3}{2}$ [4]
	fair	$\frac{1+\sqrt{17}}{4}$ [4]	$\frac{1+\sqrt{17}}{4}$ [4]	$\frac{4}{3}$ [4]	$\frac{4}{3}$ [4]
Nomadic general		≈ 2.03 $\downarrow$ (3.3)	$1+\sqrt{2}$ (4.5)	$\frac{2\sqrt{21}-3}{3}$ $\downarrow$ (3.4)	$\frac{5}{2}$ [3]
Nomadic real line	standard	≈ 2.03 (3.3)	2.06 (4.8)	$\frac{2\sqrt{21}-3}{3}$ (3.4)	$\frac{7}{3}$ [3]
	fair	$\frac{1+\sqrt{97}}{6}$ (3.5)	2.06 $\uparrow$(4.8)	$\frac{\sqrt{33}}{3}$ (3.6)	$\frac{7}{3}$ $\uparrow$ [3]
Nomadic halfline	standard	≈ 1.63 (3.7)	2.06 $\uparrow$ (4.8)	$\frac{7}{4}$ (3.9)	$\frac{7}{3}$ $\uparrow$ [3]
	fair	≈ 1.60 (3.8)	2.06 $\uparrow$ (4.8)	$\sqrt{3}$ (3.10)	$\frac{7}{3}$ $\uparrow$ [3]

Table 2.2: Lower bounds and upper bounds on the competitive ratio of deterministic algorithms for on-line dial-a ride problems.

		non-zealous		zealous	
		lower bound	upper bound	lower bound	upper bound
Homing general		2 $\leftarrow$ [3]	2 [13]	2 $\leftarrow$ [3]	$\frac{5}{2}$ [6][13]
Homing real line	standard	$\frac{7}{4}$ (3.12)	2 $\uparrow$ [13]	2 $\downarrow$ (3.11)	$\frac{5}{2}$ $\uparrow$ [6][13]
	fair	$\frac{1+\sqrt{5}}{2}$ $\downarrow$ (3.13)	2 $\uparrow$ [13]	2 $\downarrow$ (3.11)	$\frac{5}{2}$ $\uparrow$ [6][13]
Homing halfline	standard	$\frac{2+\sqrt{2}}{2}$ [13]	2 $\uparrow$ [13]	2 $\downarrow$ (3.11)	$\frac{5}{2}$ $\uparrow$ [6][13]
	fair	$\frac{1+\sqrt{5}}{2}$ (3.13)	2 $\uparrow$ [13]	2 (3.11)	$\frac{5}{2}$ $\uparrow$ [6][13]
Nomadic general		≈ 2.03 $\leftarrow$ (3.3)	$\frac{3+\sqrt{5}}{2}$ (4.9)	$\frac{5}{2}$ $\downarrow$ (3.15)	3 [13]
Nomadic real line	standard	≈ 2.03 $\leftarrow$ (3.3)	$\frac{3+\sqrt{5}}{2}$ $\uparrow$ (4.9)	$\frac{5}{2}$ $\downarrow$ (3.15)	3 $\uparrow$ [13]
	fair	$\frac{1+\sqrt{22}}{3}$ $\downarrow$ (3.14)	$\frac{3+\sqrt{5}}{2}$ $\uparrow$ (4.9)	2 $\downarrow$ (3.16)	3 $\uparrow$ [13]
Nomadic halfline	standard	$\frac{1+\sqrt{22}}{3}$ $\downarrow$ (3.14)	$\frac{3+\sqrt{5}}{2}$ $\uparrow$ (4.9)	$\frac{5}{2}$ (3.15)	3 $\uparrow$ [13]
	fair	$\frac{1+\sqrt{22}}{3}$ (3.14)	$\frac{3+\sqrt{5}}{2}$ $\uparrow$ (4.9)	2 (3.16)	3 $\uparrow$ [13]

Table 2.3: Lower bounds and upper bounds on the competitive ratio of deterministic algorithms for on-line travelling repairman problems and latency on-line dial-a ride problems.

	lower bound		upper bound	
OLTRP general	$1+\sqrt{2}$	$\downarrow$ [6]	$(1+\sqrt{2})^2$	[14]
OLTRP real line	$1+\sqrt{2}$	[6]	$(1+\sqrt{2})^2$	$\uparrow$ [14]
OLTRP halfline	2	(3.17)	$\frac{7}{2}$	[6]
L-OLDARP general	3	$\downarrow$ [6]	$(1+\sqrt{2})^2$	[14]
L-OLDARP real line	3	[6]	$(1+\sqrt{2})^2$	$\uparrow$ [14]
L-OLDARP halfline	$1+\sqrt{2}$	(3.18)	$(1+\sqrt{2})^2$	$\uparrow$ [14]

2.4 Discussion

The only open questions about deterministic algorithms for the on-line travelling salesman problem with return to the origin concerned the real line as a metric space [3], [4]. We answered those questions by designing a best possible algorithm that matches the lower bound in [3], and deriving a lower bound for the case the adversary has an imposed fairness restriction, together with designing an algorithm with matching competitive ratio. One question is still unanswered. The case with the restriction to zealous on-line algorithms against a fair adversary: the lower bound of 8/5 [19] versus the upper bound of 7/4 [3]. We believe an algorithm like WF (see Section 4.1.2) is able to close this gap. Instead of waiting until a certain time the server returns to the origin if he can reach it before a certain time.

For the on-line travelling salesman problem with a free endpoint we improved all of the existing lower bounds. We also designed an algorithm for general metric spaces and an algorithm for the real line which are currently best known. The most interesting open problem is to close the gap between the lower and the upper bound on general metric spaces: approximately 2.03 versus $1+\sqrt{2}$.

For the OLDARP, only algorithms for general metric spaces are known. No algorithm is known that applies specifically to the real line or the halfline. The only tight result for the OLDARP is for the HOLDARP in general metric spaces [14], so here also many open problems exist. The lower bounds for the HOLDARP and the NOLDARP in general metric spaces come from instances of the OLTSP. It would be interesting to see if the lower bound for the NOLDARP can be improved by using rides instead of points.

The OLTRP and L-OLDARP are analytically complex problems. We believe that the lower bound for OLTRP of $1+\sqrt{2}$ [6] is tight, even for general metric spaces. The gap with the upper bound of $(1+\sqrt{2})^2 \approx 5.83$ [14] is big. In case the lower bound of $1+\sqrt{2}$ is tight, then a best possible algorithm should, at any time t, stay within distance $(\sqrt{2}-1)t$ of the origin.

The introduction of zealousness helped us to better understand the importance of waiting in on-line routing. Our results support the strategy of waiting instead of immediately starting to serve requests. We note that every best known algorithm for general metric spaces or the real line uses an on-line server (zealous or non-zealous) who either waits in the origin, always serves the request nearest to the origin, or temporarily ignores requests until he is back in the origin. Intuitively, this makes sense because, roughly speaking, the origin is the safest place to be for an on-line server who does not known the position of the adversary.

To impose the fairness restriction on the adversary improved the competitive performance in case the metric space is the halfline or the real line. For general metric spaces it is not clear how to define the convex hull. Ausiello et al. [3] use the boundary of the unit square in the lower bound proof for HOLTSP in general metric spaces. All the points on the boundary of the unit square are presented at time 0 and the adversary is fair in the sense that he cannot go to the point of a request before its release time. The authors designed an algorithm that matches this lower bound. The lower bounds for the NOLTSP in general metric spaces are achieved on the real line and there is a difference between the fair case and the unfair case: approximately 1.81 versus approximately 2.03. It would be interesting to generalize the concept of fairness to general metric spaces. For instance, one might stipulate that the adversary is only allowed to use the shortest paths to points of requests released up till that time.

The randomized lower bounds we know are weaker than their deterministic counterparts, so we can hope for improvements on competitive performance through randomized algorithms. Krumke et al. [14] give a $4/\ln 3 \approx 3.64$-competitive randomized algorithm for the OLTRP and the L-OLDARP in general metric spaces that improves the best known deterministic competitive ratio.

Interesting extensions of on-line routing problems are to more servers, to servers with different capacities (for OLDARP), or to the weighted case in which each request gets a certain ‘importance’. Other adversary models, particular metric spaces, the restriction to polynomial time algorithms, or other objective functions are also worth studying.

3

Lower bounds for on-line routing problems

In this chapter we prove lower bounds on the competitive ratio of deterministic algorithms for on-line routing problems.

Consider any on-line server OL who is ρ-competitive. We denote the objective function value of the on-line algorithm by Z^{OL} and that of the optimal off-line objective function value by Z^*. We denote the position of the OL-server at any time t by p_t^{ol} and the position of the optimal server at time t by p_t^*.

3.1 Lower bounds for the HOLTSP

The following proof is is an alternative proof for the one presented in [3].

Theorem 3.1. *Any ρ-competitive algorithm for the HOLTSP in general metric spaces has $\rho \geq 2$.*

Proof. The metric space is a graph with vertex set $X = \{1, 2, \ldots, n\} \cup O$ and the distance function d, where $d(O, i) = 1$ and $d(i, j) = 2$ for all $i, j \in X \backslash O$.

At time 0 there is a request in each of the n points in $X \setminus \{O\}$. If OL serves the request in point i at time t with $t \leq 2n - 1 - \epsilon$, then at time $t + \epsilon$, a new request in point i is presented.

In this way, at time time $2n - 1$ OL still has to serve requests in all n points. We have $Z^{OL} \geq 2n - 1 + 2n - 1 = 4n - 2$, whereas $Z^* = 2n$, yielding ρ arbitrarily close to 2 if we take n large enough. □

Theorem 3.2. *Any ρ-competitive algorithm for the HOLTSP on the real line against a fair adversary has $\rho \geq (5+\sqrt{57})/8 \approx 1.57$.*

PROOF. At time 0, two requests $\sigma_1 = (0,-1)$ and $\sigma_2 = (0,1)$ are presented. At time 2 OL cannot have served both requests. Without loss of generality, we assume that at time 2 the position of OL is in the origin or on the negative halfline. At time 2 a request $\sigma_3 = (2,-1)$ is presented. Let t_0 denote the time at which OL returns to the origin after having served either the requests in -1 or the request in $+1$. The optimal off-line completion time $Z^* = 4$. Therefore, $t_0 \leq 4\rho - 2$. We distinguish two cases.

- If OL serves the requests at point -1 first, he cannot be back in the origin before time 3, implying $3 \leq t_0 \leq 4\rho - 2$. At t_0 a request $\sigma_4 = (t_0, -1)$ is presented. OL cannot finish before time $t_0 + 4$, whereas $Z^* = t_0 + 1$. Therefore, $\rho \geq (t_0+4)/(t_0+1)$.
- If OL serves the request at point $+1$ first, he cannot be back in the origin before time 4, implying $4 \leq t_0 \leq 4\rho - 2$. At t_0 a request $\sigma_4 = (t_0, 1)$ is presented. OL cannot finish before time $t_0 + 4$, whereas $Z^* = t_0 + 1$. Also in this case, $\rho \geq (t_0+4)/(t_0+1)$.

The ratio $(t_0+4)/(t_0+1)$ is monotonically decreasing in t_0, for $t_0 > 0$. Thus, $\rho \geq (4\rho+2)/(4\rho-1)$, implying $\rho \geq (5+\sqrt{57})/8$. □

In Section 4.1.2 we present an algorithm for this problem with a matching competitive ratio.

3.2 Lower bounds for the NOLTSP

In this section the lower bounds we found are all defined on the real line or on the halfline. Since we use the Euclidean metric on the real line, $d(v,0) = |v|$ for any point v. For notational convenience we will use v here, not only for the distance $d(v,0)$, but also to indicate the request, which actually is at point $-v$ or $+v$.

Theorem 3.3. *Any ρ-competitive algorithm for the NOLTSP in general metric spaces has $\rho > 2.02976$. The lower bound is achieved on the real line.*

PROOF. Without loss of generality, we assume that at time 1 the position of OL is in the origin, or to the left of the origin. The adversary starts the sequence at time 1 with a request in $+1$. Let x denote the time at which OL serves the request in $+1$. Clearly, $2 \leq x \leq \rho$. At x a request in point $-x$ is presented. The next request is presented at time $y > x$ in point y, such that the two following equalities hold.

$$y + d(p_y^{ol}, -x) + 2(x+y) = \alpha(2y+x) \tag{3.1}$$

and

$$y + d(p_y^{ol}, y) + 2(x+y) = \alpha(2x+y). \tag{3.2}$$

Here α is a number that depends on x and y. These two equalities are crucial in the proof and imply the underlying basic idea of the lower bound, which we explain briefly.

Suppose that after time y OL first serves $-x$. Then at time $2y+x$ a new request in point $-x$ is presented. If at time $2y+x$ OL has to serve y first, then, using (3.1), $Z^{OL} \geq \alpha(2y+x)$. At time $2y+x$ the optimal server having served y, can be in point $-x$, so $Z^* = 2y+x$. This implies that $Z^{OL}/Z^* \geq \alpha$. The case that OL first serves y after time y is symmetric, using (3.2).

We will show that, given a certain value of α, $Z^{OL}/Z^* \geq \alpha$. The value of α depends on the value of x and y. Since the value of x and y depends on the behaviour of OL, OL will behave in such a way that the value of x and y minimizes α.

Suppose $p_y^{ol} > 0$ and that at time x OL goes in the direction of $-x$ at maximum (unit) speed. Since $p_x^{ol} = 1$, we have $p_y^{ol} = x + 1 - y$. Using this, in (3.1) and (3.2) we obtain

$$\frac{4x+2y+1}{2y+x} = \alpha \tag{3.3}$$

and

$$\frac{5y+x-1}{2x+y} = \alpha. \tag{3.4}$$

Hence,

$$8y^2 - y(x+3) - 7x^2 - 3x = 0,$$

$$y = \frac{3+x+\sqrt{225x^2+102x+9}}{16},$$

and

$$\alpha = \frac{11+33x+\sqrt{225x^2+102x+9}}{3+9x+\sqrt{225x^2+102x+9}}. \tag{3.5}$$

Since $2 \leq x \leq \rho$, we have

$$\frac{11+33\rho+\sqrt{225\rho^2+102\rho+9}}{3+9\rho+\sqrt{225\rho^2+102\rho+9}} \leq \alpha \leq \frac{77+\sqrt{1113}}{21+\sqrt{1113}} \approx 2.030.$$

α is minimal, hence equal to ρ, if $x = \rho$. Using $\alpha = \rho = x$ in (3.3) and (3.4) yields

$$y = \frac{x^2-4x-1}{2-2x}$$

and

$$y = \frac{2x^2 - x + 1}{5 - x}.$$

If we combine these equalities we obtain

$$3x^3 + 3x^2 - 15x - 7 = 0. \tag{3.6}$$

The solution of this equality is the minimum value for α and hence equal to ρ. Now we will show that for all possible values of α, as computed by (3.5), the completion time of OL is at least α times the optimal off-line solution. We note that any value of α only depends on x, since we assumed that OL goes straight to point $-x$ after time x.

If we assume again that $p_y^{ol} > 0$ then according to (3.1)

$$p_y^{ol} = \alpha(2y + x) - 3x - 3y$$

and according to (3.2)

$$p_y^{ol} = 4y + 2x - \alpha(2x + y).$$

If we make α only depending on x, then for fixed x, these equalities are linear functions of y. Thus, given x, we obtain two lines l_1 and l_2 with

$$l_1(y) = \alpha(2y + x) - 3x - 3y = (2\alpha - 3)y + (\alpha - 3)x \tag{3.7}$$

and

$$l_2(y) = 4y + 2x - \alpha(2x + y) = (4 - \alpha)y + (2 - 2\alpha)x. \tag{3.8}$$

If at time x OL goes in the direction of $-x$ at maximum (unit) speed, then by construction at time y OL is in the point where l_1 and l_2 cross. If at time x OL does not go in the direction of $-x$ at maximum (unit) speed then OL crosses l_2 at a certain point before crossing l_1. This is true because $4 - \alpha > 2\alpha - 3 \geq 1$ for $7/3 \geq \alpha \geq 2$. To make the analysis easier we assume y is presented the moment OL crosses l_2. At this time equality (3.2) still holds, while equality (3.1) becomes inequality

$$y + d(p_y^{ol}, -x) + 2(x + y) \geq \alpha(2y + x). \tag{3.9}$$

If we use the minimum value for α, as implied by (3.6), for all choices of x, then OL may cross l_1 before crossing l_2 thereby violating (3.2). We note that OL crosses l_2 at a point to the right of the origin, hence $p_y^{ol} > 0$ in all cases. If OL keeps going to the right after time x, then $p_y^{ol} = y - x + 1$ and OL crosses l_2 as far to the right as possible.

We have that y is presented the moment that $y-x+1=(4-\alpha)y+(2-2\alpha)x$. Thus, the maximum value of y for a certain value of x and α (y_x^{max}), is

$$y_x^{max}=\frac{(2\alpha-3)x+1}{3-\alpha}. \qquad (3.10)$$

We note that y_x^{max} is maximal and approximately 3.26 if $x=\rho$.

Summarizing, we have the following request sequence. At time 1 a request in point $+1$ is given. When, at time x OL serves this request, a request in point $-x$ is given. At time y, when $p_y^{ol}=4y+2x-\alpha(2x+y)$, with α calculated by (3.5), a request in point $+y$ is presented. We will show that for all choices of x, $Z^{OL}/Z^*\geq\alpha$. This implies that the minimal value of α, given in (3.6), equals ρ.

If OL serves $-x$ before y, then we define W_x as the time OL waited or lingered between y and $t=2y+x+2/3W_x$. If no further requests are given, then $Z^{OL}\geq 2y+2x+p_y^{ol}+W_x=6y+4x+W_x-\alpha(2x+y)$ and $Z^*=2x+y$. Thus, for OL to be α-competitive

$$W_x\leq 2\alpha(2x+y)-6y-4x. \qquad (3.11)$$

If OL serves y before $-x$, then we define W_y as the time OL waited or lingered between y and $t=2x+y+2/3W_y$. If no further requests are given, then $Z^{OL}\geq 3y+x-p_y^{ol}+W_y=\alpha(2x+y)-y-x$. Thus, for OL to be α-competitive

$$W_y\leq x+y. \qquad (3.12)$$

We distinguish two situations.

- *OL serves $-x$ before y.*
  At time $t_q=2y+x+2/3W_x$ a request in point $q=-x-2/3W_x$ is presented. If $y\geq p_y^{ol}+1/3W_x$ for all W_x, then $2y+x+2/3W_x\geq y+x+p_y^{ol}+W_x$, and at t_q OL has served $-x$. To show $y\geq p_y^{ol}+1/3W_x$, we use $x\geq 2$, $\alpha\geq 2$, and $y\leq\approx 3.26$, implying that $\alpha(2x+y)>3y+2x$. Then, using (3.11), we obtain

$$\begin{aligned} y &> y+y+\frac{2x}{3}-\frac{\alpha(2x+y)}{3} \\ &= 4y+2x-\alpha(2x+y)+\frac{2\alpha(2x+y)-6y-4x}{3} \\ &\geq p_y^{ol}+1/3W_x. \end{aligned}$$

  If $W_x=0$, then OL reaches $-x$ at time $y+x+p_y^{ol}$. Hence, at t_q OL is at distance $y-p_y^{ol}$ to the right from q. If $W_x>0$, then at t_q OL is $1/3W_x$ to the left of the position in which OL would be if $W_x=0$ and $d(p_{t_q}^{ol},q)=y-p_y^{ol}+1/3W_x$. Again we distinguish two situations.

  - $d(p_{t_q}^{ol},q)\geq d(p_{t_q}^{ol},y)$.
    Using (3.9) we have that $Z^{OL}\geq\alpha(2y+x)+5/3W_x$. The optimal server,

having served y, can be in q at time t_q, so $Z^* = 2y + x + 2/3W_x$. Clearly, $Z^{OL}/Z^* \geq \alpha$.

– $d(p^{ol}_{t_q}, q) < d(p^{ol}_{t_q}, y)$.
The optimal server keeps going to the left after time t_q. At some time OL has to start going in the direction of y. A request in point $z = -q - s$ is presented at time $t_m = t_q + s$ with $s > 0$, at which $d(p^{ol}_{t_m}, z) = d(p^{ol}_{t_m}, y)$. Using (3.9) we have that $Z^{OL} > \alpha(2y + x) + 5/3W_x + 5/2s$, whereas $Z^* = 2y + x + 2/3W_x + s$. Clearly, $Z^{OL}/Z^* \geq \alpha$.

- *OL serves y before $-x$.*
At time $t_v = 2x + y + 2/3W_y$ a request in point $v = y + 2/3W_y$ is presented. If $2x \geq y - p^{ol}_y + 1/3W_y$ for all W_y, then $2x + y + 2/3W_y \geq 2y - p^{ol}_y + W_y$ and at t_v OL has served y. To show $2x \geq y - p^{ol}_y + 1/3W_y$, we use $x < y$ and $\alpha \leq\approx 2.03$ implying that $3\alpha(2x + y) < 8y + 11x$. Then, using (3.12), we obtain

$$\begin{aligned} 2x &> 2x + \alpha(2x + y) - 8/3y - 11/3x \\ &= \alpha(2x + y) - 8/3y - 5/3x \\ &\geq y - p^{ol}_y + 1/3W_y. \end{aligned}$$

If $W_y = 0$, then OL reaches y at time $2y - p^{ol}_y$, so at t_v OL is at distance $2x - y + p^{ol}_y$ to the left from v. If $W_y > 0$, then at t_v OL is $1/3W_y$ to the right of the position in which OL would be if $W_y = 0$ and $d(p^{ol}_{t_v}, v) = 2x - y + p^{ol}_y + 1/3W_y$. We distinguish two situations.

– $d(p^{ol}_{t_v}, -x) \leq d(p^{ol}_{t_v}, v)$.
Using (3.2) we have that $Z^{OL} \geq \alpha(2x+y) + 5/3W_y$. The optimal server, having served y, can be in v at time t_v, so $Z^* = 2x + y + 2/3W_y$. Clearly, $Z^{OL}/Z^* \geq \alpha$.

– $d(p^{ol}_{t_v}, -x) > d(p^{ol}_{t_v}, v)$.
The optimal server keeps going to the right after time t_v. At some time OL has to start going in the direction of $-x$. A request in point $z' = v + s'$ is presented at time $t'_m = t_v + s'$ with $s' > 0$, at which $d(p^{ol}_{t'_m}, z') = d(p^{ol}_{t'_m}, -x)$. Using (3.2) we have that $Z^{OL} > \alpha(2x + y) + 5/3W_y + 5/2s'$, whereas $Z^* = 2x + y + 2/3W_y + s'$. Clearly, $Z^{OL}/Z^* \geq \alpha$.

□

Theorem 3.4. *Any zealous ρ-competitive algorithm for the NOLTSP in general metric spaces has $\rho \geq (2\sqrt{21} - 3)/3 \approx 2.06$. The lower bound is achieved on the real line.*

Proof. At time 0 a request $\sigma_1 = (0, 1)$ is presented. At time 1 a request $\sigma_2 = (1, -1)$ and at time $t_1 = (1 + \sqrt{21})/4 \approx 1.40$ a request $\sigma_3 = (t_1, t_1)$ is presented. At time t_1 any zealous server must be in point $2 - t_1$. We distinguish three situations.

- *OL goes in the direction of σ_2 and does not turn around in the origin.*
  At time $2t_1+1$, a request $\sigma_4 = (2t_1+1, -1)$ is presented. At this time OL must be in point $2t_1 - 3$, whereas the adversary is in point -1. If OL serves σ_3 first, then OL cannot finish before time $5 + 2t_1 = (11 + \sqrt{21})/2$, whereas $Z^* = 2t_1 + 1 = (3 + \sqrt{21})/2$. Therefore, $Z^{OL}/Z^* \geq (2\sqrt{21} - 3)/3$.

  If OL starts serving σ_4 at time $2t_1 + 1$, then at time $t_5 = 11t_1/3$ a request $\sigma_5 = (t_5, 2t_1 - t_5)$ is presented. At time t_5 OL is in point $-t_1/3$, whereas the adversary is in point $2t_1 - t_5 = -5/3t_1$. OL cannot finish before time $11t_1/3 + t_1/3 + 2t_1 + 5t_1/3 = 23t_1/3$, whereas $Z^* = t_5 = 11t_1/3$. Therefore, $Z^{OL}/Z^* \geq 23/11 > (2\sqrt{21} - 3)/3$.

  If OL goes back to the origin at time $2t_1+1$ and then starts serving σ_4, then at time $t_5 = 11t_1/3+2p^{ol}_{2+t_1}/3$ a request $\sigma_5 = (t_5, -5t_1/3-2p^{ol}_{2+t_1}/3)$ is presented. At time t_5 OL is in point $-t_1/3 - p^{ol}_{2+t_1}/3$, whereas the adversary is in point $-5t_1/3 - 2p^{ol}_{2+t_1}/3$. OL cannot finish before time $23t_1/3 + 5p^{ol}_{2+t_1}/3$, whereas $Z^* = t_5 = 11t_1/3 + 2p^{ol}_{2+t_1}/3$. Therefore, $Z^{OL}/Z^* > 23/11 > (2\sqrt{21} - 3)/3$.

- *OL goes in the direction of σ_3.*
  At time $2 + t_1$, a request $\sigma_4 = (2 + t_1, t_1)$ is presented. At time $2 + t_1$ OL must be in point $3t_1 - 4$, whereas the adversary is in point t_1. Since $d(p^{ol}_{2+t_1}, -1) < d(p^{ol}_{2+t_1}, t_1)$ it suffices to consider the case in which after time $2 + t_1$ OL serves σ_2 first. OL cannot finish before time $5t_1 = (5 + 5\sqrt{21})/4$, whereas $Z^* = 2 + t_1 = (9 + \sqrt{21})/4$. Therefore, $Z^{OL}/Z^* \geq (2\sqrt{21} - 3)/3$.

- *OL goes back to the origin and then starts serving σ_3.*
  At time $t_4 = (7+4t_1)/3 = (8+\sqrt{21})/3$ a request $\sigma_4 = (t_4, t_4 - 2)$ is presented. At time t_4 OL is in point $(2t_1 - 1)/3$, whereas the adversary is in point $t_4 - 2$. Since $d(p^{ol}_{t_4}, -1) < d(p^{ol}_{t_4}, t_4-2)$ OL cannot finish before time $2+2t_1+t_4 = (31+5\sqrt{21})/5$, whereas $Z^* = t_4 = (8+\sqrt{21})/3$. Therefore, $Z^{OL}/Z^* > (2\sqrt{21}-3)/3$.

$\square$

Theorem 3.5. *Any ρ-competitive algorithm for the NOLTSP on the real line against a fair adversary has $\rho \geq (1 + \sqrt{97})/6 \approx 1.81$.*

PROOF. At time 0, two requests $\sigma_1 = (0, 1)$ and $\sigma_2 = (0, -1)$ are presented. Without loss of generality we assume $p^{ol}_1 \geq 0$. Then, $\sigma_3 = (1, y)$ with $y = (1+\sqrt{97})/8) \approx 1.35$ is presented. We distinguish three situations at time $t' = 2+y$.

- *At time t' OL has served σ_2.*
  If, at time $t_4 = 2y + 1$, $d(p^{ol}_{t_4}, -1) \geq d(p^{ol}_{t_4}, y)$, then a request $\sigma_4 = (t_4, -1)$ is presented. OL cannot finish before time $4 + 2y = (17 + \sqrt{97})/4$, whereas $Z^* = 2y + 1 = (5 + \sqrt{97})/4$. Thus, $Z^{OL}/Z^* \geq (1 + \sqrt{97})/6$.

  If $d(p^{ol}_{t_4}, -1) < d(p^{ol}_{t_4}, y)$, then at time t_4 the adversary starts giving requests at distance ϵ to the left of his own position. If OL continues to serve these requests before the request in point y, then he will have a competitive ratio of 2. Thus, at a certain time OL has to start going in the direction of the request in point y. The adversary continues giving requests at distance ϵ to his left until time t_m at which the distance between OL and the adversary equals the distance between OL and point y. OL cannot finish before time $t_m + 3/2(t_m - y)$, whereas $Z^* = t_m + \epsilon$. The ratio $(5/2t_m - 3/2y)/(t_m + \epsilon)$ is monotonically increasing in t_m, for $t_m > 2y + 1$, implying $Z^{OL}/Z^* > (1 + \sqrt{97})/6$.

- *At time t' OL has served σ_3.*
  If $d(p^{ol}_{t'}, -1) \leq d(p^{ol}_{t'}, y)$, then a request $\sigma_4 = (t', y)$ is presented. OL cannot finish before time $3y+2 = (19+3\sqrt{97})/8$, whereas $Z^* = 2+y = (17+\sqrt{97})/8$. Therefore, $Z^{OL}/Z^* \geq (1 + \sqrt{97})/6$.

  If $d(p^{ol}_{t'}, -1) > d(p^{ol}_{t'}, y)$, then at time t' the adversary starts giving requests at distance ϵ to the right of his own position until time t_m at which the distance between OL and the adversary equals the distance between OL and point -1. OL cannot finish before time $t_m + 3/2(t_m - 1)$, whereas $Z^* = t_m + \epsilon$. The ratio $(5/2t_m - 3/2)/(t_m + \epsilon)$ is monotonically increasing in t_m, for $t_m > 2 + y$, implying $Z^{OL}/Z^* > (1 + \sqrt{97})/6$.

- *At time t' OL has neither served σ_2 nor σ_3.*
  If $d(p^{ol}_{t'}, -1) > d(p^{ol}_{t'}, y)$, then at time t' the adversary starts giving requests at distance ϵ to the right of his own position until time t_m at which the distance between OL and the adversary equals the distance between OL and point -1. OL cannot finish before time $t_m + 3/2(t_m - 1)$, whereas $Z^* = t_m + \epsilon$. The ratio $(5/2t_m - 3/2)/(t_m + \epsilon)$ is monotonically increasing in t_m, for $t_m > 2 + y$, implying $Z^{OL}/Z^* > (1 + \sqrt{97})/6$.

  If $d(p^{ol}_{t'}, -1) \leq d(p^{ol}_{t'}, y)$, then at time $t_4 = 2y + 1$ OL cannot have served σ_3. If $d(p^{ol}_{t_4}, -1) \geq d(p^{ol}_{t_4}, y)$, then a request $\sigma_4 = (t_4, -1)$ is presented. OL cannot finish before time $2 + y + (1 + y)/2 + y + 1 = (61 + 5\sqrt{97})/16$, whereas $Z^* = 2y + 1 = (5 + \sqrt{97})/4$. Thus, $Z^{OL}/Z^* \geq (1 + \sqrt{97})/6$.

  If $d(p^{ol}_{t_4}, -1) < d(p^{ol}_{t_4}, y)$, then at time t_4 the adversary starts giving requests at distance ϵ to the left of his own position. The adversary continues giving requests at distance ϵ to his left until time t_m at which the distance between OL and the adversary equals the distance between OL and point y. OL cannot finish before time $t_m + 3/2(t_m - y)$, whereas $Z^* = t_m + \epsilon$. The ratio $(5/2t_m - 3y/2)/(t_m + \epsilon)$ is monotonically increasing in t_m, for $t_m > 2y + 1$, implying $Z^{OL}/Z^* > (1 + \sqrt{97})/6$.

□

Theorem 3.6. *Any zealous ρ-competitive algorithm for the NOLTSP on the real line against a fair adversary has $\rho \geq \sqrt{33}/3 \approx 1.91$.*

PROOF. At time 0, two requests $\sigma_1 = (0, 1)$ and $\sigma_2 = (0, -1)$ are presented. Without loss of generality we assume OL starts serving σ_1. Then a request $\sigma_3 = (1, y)$ with $y = (1+\sqrt{33})/4 \approx 1.69$ is presented. At this time any zealous algorithm must be in point $+1$. We distinguish three situations.

- *OL goes in the direction of σ_2 and does not turn around in the origin.*
  At time $2y+1$, a request $\sigma_4 = (2y+1, -1)$ is presented. At this time OL must be in point $2y-3$, whereas the adversary is in point -1. Since $d(p^{ol}_{2y+1}, -1) > d(p^{ol}_{2y+1}, y)$ it suffices to consider the case in which after time $2y+1$ OL serves σ_3. In this case OL cannot finish before time $4 + 2y + 1 = (22 + 2\sqrt{33})/4$, whereas $Z^* = 2y+1 = (6+2\sqrt{33})/4$. Therefore, $Z^{OL}/Z^* \geq (22+2\sqrt{33})/(6+2\sqrt{33}) = \sqrt{33}/3$.

- *OL goes in the direction of σ_3 at time* 1.
  At time $2+y$, a request $\sigma_4 = (2+y, y)$ is presented. At this time OL is in point $y - 2$, whereas the adversary is in point y. Since $d(p^{ol}_{2+y}, -1) \geq d(p^{ol}_{2+y}, y)$ it suffices to consider the case in which after time $2 + y$ OL serves σ_2. Then, OL cannot finish before time $2y + 2 + y = (11 + 3\sqrt{33})/4$, whereas $Z^* = 2 + y = (9 + \sqrt{33})/4$. Therefore, $Z^{OL}/Z^* \geq (11 + 3\sqrt{33})/(9 + \sqrt{33}) = \sqrt{33}/3$.

- *OL goes to the origin at time* 1*. In the origin OL turns around and starts serving σ_3.*
  At time $2 + y$, both OL and the adversary are in point y. At time $2 + y$ the adversary starts giving requests at distance ϵ to his right. If OL continues to serve these requests before the request in point -1, then he will have a competitive ratio of 2. Thus, at a certain time OL has to start going in the direction of the request in point -1. The adversary continues giving requests at distance ϵ to his right until time t_m at which the distance between OL and the adversary equals the distance between OL and point -1. OL cannot finish before time $t_m + 3/2(t_m - 1)$, whereas $Z^* = t_m + \epsilon$. The ratio $(5/2t_m - 3/2)/(t_m + \epsilon)$ is monotonically increasing in t_m, for $t_m > 0$. The lowest possible value of t_m is $(8 + \sqrt{33})/3$, implying $Z^{OL}/Z^* \geq (31 + 5\sqrt{33})/(16 + 2\sqrt{33}) > \sqrt{33}/3$.

□

Theorem 3.7. *Any ρ-competitive algorithm for the NOLTSP on the halfline has $\rho \geq\approx 1.63$.*

PROOF. At time 1, a request $\sigma_1 = (1, 1)$ is presented. Let t_1 denote the time at which OL serves request σ_1. Clearly, $1 \leq t_1 \leq \rho$. At time t_1 two requests $\sigma_2 = (t_1, 0)$ and $\sigma_3 = (t_1, b)$ are presented. Here, b is a fixed number for which the two following equalities hold:

$$t_1 + 1 + 2b = \alpha(2b) \tag{3.13}$$

and

$$\rho(t_1 + b) + b = \alpha(\rho(t_1 + b) - b/2). \tag{3.14}$$

Here, α is a fixed number whose value depends on b. We will show that $Z^{OL}/Z^* \geq \alpha$ for any value of α. The minimum value of α equals ρ and is attained when $t_1 = \rho$. If we solve the equalities (3.13) and (3.14) using $t_1 = \rho$, then we get for the value b the solution of $-12b^4 + 22b^3 + 3b^2 + 13b/2 + 2 = 0$, which is approximately 2.10. For α we get the solution of $4\alpha^4 - 6\alpha^3 + 3\alpha^2 - 5\alpha - 2 = 0$, which is approximately 1.63. We note that b is minimal and approximately 1.49 when $t_1 = 1$.

The first equality implies that OL is α-competitive in case he follows a greedy tour first serving the request in point 0, then the request in point b and, finally, a new request in point 0 again. If OL follows a tour first serving the request in point 0 waiting as long as possible near, or in the origin, then a new request in point 0 is presented at time t_m such that $d(p^{ol}_{t_m}, 0) = d(p^{ol}_{t_m}, b)$. The second equality implies that OL is α-competitive for all possible t_m.

We distinguish two situations after time t_1.

- *OL serves request σ_2 first.*
  OL has to serve request σ_2 at some time $t_0 \leq \rho(t_1 + b) - b$. If $t_0 \leq 2b$ and $d(p^{ol}_{2b}, 0) \geq d(p^{ol}_{2b}, b)$, then a request $\sigma_4 = (2b, 0)$ is presented. Using (3.13), we have that $Z^{OL}/Z^* \geq \alpha$. In all other cases the adversary presents the next request in point 0 at time $t_m > t_0$ at which $d(p^{ol}_{t_m}, 0) = d(p^{ol}_{t_m}, b)$. OL must serve the request in point b before time $\rho(t_1 + b)$, since $Z^* \leq t_1 + b$. Therefore, t_m must occur and $t_m \leq \rho(t_1 + b) - b/2$. OL cannot finish before time $t_m + 3/2b$, whereas $Z^* = t_m$. The ratio $(t_m + 3/2b)/t_m$ is monotonically decreasing in t_m, for $t_m > t_0$. Using equality (3.14), we have that $Z^{OL}/Z^* \geq \alpha$.

- *OL serves request σ_3 first.*
  OL has to serve request σ_3 at some time $t_b \leq \rho(t_1 + b) - b$. If $t_b \leq t_1 + b$ and $d(p^{ol}_{t_1+b}, 0) \leq d(p^{ol}_{t_1+b}, b)$, then a request $\sigma_4 = (t_1 + b, b)$ is presented. OL cannot finish before time $t_1 + 3b - 1$, whereas $Z^* = t_1 + b$. Using $b \geq\approx 1.49$, we have $Z^{OL}/Z^* \geq (t_1 + 3b - 1)/(t_1 + b) > \rho$.

  In all other cases the adversary continues going to the right until time t_m at which the distance between OL and the adversary equals the distance between OL and point 0. Then a request in point $t_m - t_1$ is presented. OL cannot finish before time $t_m + 3/2(t_m - t_1)$, whereas $Z^* = t_m$. The ratio $(5/2t_m - 3/2t_1)/t_m$ is monotonically increasing in t_m, for $t_m > t_1 + b$, implying $Z^{OL}/Z^* > (t_1 + 3b - 1)/(t_1 + b) > \rho$.

□

Theorem 3.8. *Any ρ-competitive algorithm for the NOLTSP on the halfline against a fair adversary has $\rho \geq\approx 1.60$.*

PROOF. At time 0, a request $\sigma_1 = (0, 1)$ is presented. At time 1 we have that $2 - \rho \leq p_1^{ol} \leq 1$. At time 1 two requests $\sigma_2 = (1, 0)$ and $\sigma_3 = (1, b)$ are presented. Here, b is a fixed number for which the two following equalities hold:

$$1 + p_1^{ol} + 2b = \alpha(2b) \tag{3.15}$$

and

$$\rho(1 + b) + b = \alpha(\rho(1 + b) - b/2). \tag{3.16}$$

Here, α is a fixed number whose value depends on b. We will show that $Z^{OL}/Z^* \geq \alpha$ for any value of α. The minimum value of α equals ρ and is attained when OL is in point $2 - \rho$ at time 1. If we solve these equalities with $\rho = \alpha$ and $p_1^{ol} = 2 - \rho$, then we get for the value b the solution of $12b^3 + 8b^2 - 15b - 12 = 0$, which is approximately 1.16. For α we get the solution of $2\alpha^3 + \alpha^2 - 3\alpha - 6 = 0$, which is approximately 1.60. We note that b is minimal and approximately 1.16 when $p_1^{ol} = 2 - \rho$.

The first equality implies that OL is α-competitive in case he follows a greedy tour first serving the request in point 0, then the request in point b and, finally, a new request in point 0 again. If OL follows a tour first serving the request in point 0 waiting as long as possible near, or in the origin, then a new request in point 0 is presented at time $t_m \leq \rho(t_1 + b) - b/2$ such that $d(p_{t_m}^{ol}, 0) = d(p_{t_m}^{ol}, b)$. The second equality implies that OL is α-competitive for all possible t_m.

We distinguish two situations at time $t_q = 1 + b$.

- $d(p_{t_q}^{ol}, 0) \geq d(p_{t_q}^{ol}, b)$.
  If OL did not serve σ_2 yet, then the adversary starts giving requests at distance ϵ to the right of his own position. At a certain time OL has to start going in the direction of the request in point 0. The adversary continues giving requests at distance ϵ to his right until time t_m at which the distance between OL and the adversary equals the distance between OL and point 0. OL cannot finish before time $t_m + 3/2(t_m - 1)$, whereas $Z^* = t_m + \epsilon$. The ratio $(5/2t_m - 3/2)/(t_m + \epsilon)$ is monotonically increasing in t_m, for $t_m > 1 + b$. Using $b \geq\approx 1.16$, we have $Z^{OL}/Z^* \geq 3b/(1 + b) > \rho$.

  If OL has served σ_2 at t_q, then a request $\sigma_4 = (t_q, 0)$ is presented. Using equality (3.15), we have that $Z^{OL}/Z^* \geq \alpha$.

- $d(p_{t_q}^{ol}, 0) < d(p_{t_q}^{ol}, b)$.
  If OL has served σ_2 at t_q, then the adversary presents the next request in point 0 at time $t_m > t_0$ at which $d(p_{t_m}^{ol}, 0) = d(p_{t_m}^{ol}, b)$. OL must serve the request in point b before time $\rho(1 + b)$, since $Z^* \leq 1 + b$. Therefore, t_m must occur and $t_m \leq \rho(1 + b) - b/2$. OL cannot finish before time $t_m + 3/2b$, whereas $Z^* = t_m$. The ratio $(t_m + 3/2b)/t_m$ is monotonically decreasing in t_m, for $t_m > t_0$. Using equality (3.16), we have that $Z^{OL}/Z^* \geq \alpha$.

If OL has not served σ_2 at t_q, but has served σ_3 then a request $\sigma_4 = (t_q, b)$ is presented. OL cannot finish before time $3b$, whereas $Z^* = 1 + b$. Using $b \geq\approx 1.16$, we have $Z^{OL}/Z^* \geq 3b/(1+b) > \rho$.

Otherwise, OL has neither served σ_2 nor σ_3 at t_q. If after t_q OL serves σ_3 before σ_2, then OL cannot finish before time $1 + 5/2b$, whereas $Z^* = 1 + b$. Using $b\ geq \approx 1.16$, we have $Z^{OL}/Z^* \geq (1 + 5/2b)/(1 + b) > \rho$.

If OL serves σ_3 before σ_2, then the adversary presents the next request in point 0 at time $t_m > t_0$ at which $d(p^{ol}_{t_m}, 0) = d(p^{ol}_{t_m}, b)$. OL must serve the request in point b before time $\rho(1 + b)$, since $Z^* \leq 1 + b$. Therefore, t_m must occur and $t_m \leq \rho(1+b)-b/2$. OL cannot finish before time $t_m+3/2b$, whereas $Z^* = t_m$. Using equality (3.16), we have that $Z^{OL}/Z^* \geq \alpha$.

□

Theorem 3.9. *Any zealous ρ-competitive algorithm for the NOLTSP on the halfline has $\rho \geq 7/4$.*

Proof. At time 0, a request $\sigma_1 = (0, 3/2)$ is presented. At time 3/2 a request $\sigma_2 = (3/2, 1)$ is presented. At time 2, $p^{ol}_2 = 1$ and two requests $\sigma_3 = (2, 0)$ and $\sigma_4 = (2, 2)$ are presented. If OL serves σ_3 first, then, at time 4, $p^{ol}_4 = 1$ and a request $\sigma_5 = (4, 0)$ is presented.

If OL serves σ_4 first, then at time 4, $p^{ol}_4 = 1$ and a request $\sigma_5 = (4, 2)$ is presented. In both cases OL cannot finish before time 7, whereas $Z^* = 4$. Therefore, $\rho \geq 7/4$. □

Theorem 3.10. *Any zealous ρ-competitive algorithm for the NOLTSP on the halfline against a fair adversary has $\rho \geq \sqrt{3} \approx 1.73$.*

Proof. At time 0, a request $\sigma_1 = (0, 1)$ is presented. At time 1 two requests $\sigma_2 = (1, 0)$ and $\sigma_3 = (1, (1 + \sqrt{3})/2)$ are presented. If OL serves σ_2 first, then $p^{ol}_{1+\sqrt{3}} = \sqrt{3}-1$ and a request $\sigma_4 = (1+\sqrt{3}, 0)$ is presented. OL cannot finish before time $3+\sqrt{3}$, whereas $Z^* = 1+\sqrt{3}$. Therefore, $\rho \geq (3+\sqrt{3})/(1+\sqrt{3}) = \sqrt{3}$.

If OL serves σ_3 first, then $p^{ol}_{(3+\sqrt{3})/2} = (\sqrt{3} - 1)/2$ and a request $\sigma_4 = ((3 + \sqrt{3})/2, (1+\sqrt{3})/2))$ is presented. OL cannot finish before time $(3+3\sqrt{3})/2$, whereas $Z^* = (3 + \sqrt{3})/2$. Therefore, $\rho \geq (3 + 3\sqrt{3})/(3 + \sqrt{3}) = \sqrt{3}$. □

3.3 Lower bounds for the HOLDARP

Theorem 3.11. *Any zealous ρ-competitive algorithm for the HOLDARP in general metric spaces against a fair adversary has $\rho \geq 2$. The lower bound is achieved on the halfline.*

PROOF. At time 0, two requests $\sigma_1 = (0, 0, 1)$ and $\sigma_2 = (0, 1, 0)$ are presented. At time 1 OL picks up ride σ_2. At time 1 a request $\sigma_3 = (1, 1, 1)$ is presented. We have $Z^{OL} = 4$ and $Z^* = 2$, yielding $\rho \geq 2$. □

Theorem 3.12. *Any ρ-competitive algorithm for the HOLDARP on the real line has $\rho \geq 7/4$.*

PROOF. Before time 1 no requests are presented. At time 1, a request $\sigma_0 = (1, 0, 0)$ is presented. Clearly, OL has to serve this request at some time $t_0 \leq \rho$. Then two requests $\sigma_1 = (t_0, t_0, 0)$ and $\sigma_2 = (t_0, -t_0, 0)$ are presented. Without loss of generality, we assume that OL serves ride σ_1 first. Let t_1 denote the time at which OL picks up ride σ_1 in point $+t_0$. The optimal off-line completion time for these requests is $4t_0$. Therefore, $2t_0 \leq t_1 \leq 4\rho t_0 - 3t_0$. At time t_1 request $\sigma_3 = (t_1, t_0, t_0)$ is presented. OL cannot finish before time $t_1 + 5t_0$. If $2t_0 \leq t_1 \leq 4t_0$, then $Z^* = 4t_0$, and $\rho \geq (t_1 + 5t_0)/4t_0$. If $3t_0 \leq t_1 \leq 4\rho t_0 - 3t_0$, then $Z^* = t_1 + t_0$, and $\rho \geq (t_1 + 5t_0)/(t_1 + t_0)$.

The ratio $(t_1+5t_0)/(t_1+t_0)$ is monotonically decreasing in t_1 for $t_1 > 0$, implying $\rho \geq 7/4$. □

Theorem 3.13. *Any ρ-competitive algorithm for the HOLDARP on the halfline against a fair adversary has $\rho \geq (1 + \sqrt{5})/2 \approx 1.62$.*

PROOF. At time 0, a request $\sigma_1 = (0, 1, 0)$ is presented. Clearly, OL has to pick up ride σ_1 at some time $t_1 \leq 2\rho - 1$. Then request $\sigma_2 = (t_1, 1, 1)$ is presented. OL cannot finish before time t_1+3, whereas $Z^* = t_1+1$. Therefore, $\rho \geq (t_1+3)/(t_1+1)$. The ratio $(t_1 + 3)/(t_1 + 1)$ is monotonically decreasing in t_1 for $t_1 > 0$, implying $\rho \geq (1 + \sqrt{5})/2$. □

3.4 Lower bounds for the NOLDARP

Theorem 3.14. *Any ρ-competitive algorithm for the NOLDARP on the halfline against a fair adversary has $\rho \geq (1 + \sqrt{22})/3 \approx 1.90$.*

PROOF. At time 0, a request $\sigma_1 = (0, 0, 1)$ is presented. Clearly, OL has to pick up ride σ_1 at some time $t_1 \leq \rho - 1$. Then, a request $\sigma_2 = (t_1, 0, 0)$ is presented. At time $t_1 + 1$ the adversary is in point 1, having served both σ_1 and σ_2. He starts giving rides of length 0 at distance ϵ to his right. OL can start serving these requests, but at a certain time he has to turn around to serve σ_2. The adversary continues giving requests at distance ϵ to his right until time t_m at which the distance between OL and the adversary equals the distance between OL and point 0. The last request is in point $t_m - t_1 + \epsilon$ and the adversary is in point $t_m - t_1$. OL cannot finish before time $t_m + 3/2(t_m - t_1 + \epsilon)$, whereas $Z^* = t_m + \epsilon$. The ratio $(t_m + 3/2(t_m - t_1 + \epsilon))/(t_m + \epsilon)$

is monotonically decreasing in t_1, for $t_1 > 0$, and monotonically increasing in t_m, for $t_m > 0$, implying $\rho \geq (1 + \sqrt{22})/3$. $\square$

Theorem 3.15. *Any zealous ρ-competitive algorithm for the NOLDARP on the halfline has $\rho \geq 5/2$.*

Proof. At time 0, a request $\sigma_1 = (0, 0, 1)$ is presented, and at time ϵ, a request $\sigma_2 = (\epsilon, 0, 0)$ is presented. At time 4/3 a request $\sigma_3 = (4/3, 4/3, 4/3)$ is presented. OL is in point 2/3 at time 4/3, so we have $Z^{OL} = 10/3$ and $Z^* = 4/3 + \epsilon$, yielding $\rho \geq 5/2$. $\square$

Theorem 3.16. *Any zealous ρ-competitive algorithm for the NOLDARP on the halfline against a fair adversary has $\rho \geq 2$.*

Proof. At time 0, a request $\sigma_1 = (0, 0, 1)$ is presented and at time ϵ, a request $\sigma_2 = (\epsilon, 0, 0)$ is presented. We have $Z^{OL} = 2$ and $Z^* = 1 + \epsilon$, yielding $\rho \geq 2$. $\square$

3.5 Lower bounds for the OLTRP and the L-OLDARP

Theorem 3.17. *Any ρ-competitive algorithm for the OLTRP on the halfline has $\rho \geq 2$.*

Proof. Before time 1 no requests are presented. At time 1, a request $\sigma_1 = (1, 0)$ is presented. Clearly, OL has to serve this request at some time $t_0 \leq \rho$. Then at time t_0 n requests in point t_0 are presented. We have $Z^{OL} = t_0 + n2t_0$ and $Z^* = nt_0 + 2t_0$, yielding ρ arbitrarily close to 2 if we take n large enough. $\square$

Theorem 3.18. *Any ρ-competitive algorithm for the L-OLDARP on the halfline has $\rho \geq 1 + \sqrt{2} \approx 2.41$.*

Proof. At time 0, a request $\sigma_1 = (0, 0, 1)$ is presented. Let t_1 denote the time at which OL picks up ride σ_1. Clearly, $t_1 \leq \rho - 1$. At time t_1 n requests $\sigma_n = (t_1, 0, 0)$ are presented. We have $Z^{OL} = t_1 + n(2 + t_1)$ and $Z^* = nt_1 + t_1 + 1$. The ratio $t_1 + n(2 + t_1)/nt_1 + t_1 + 1$ is monotonically decreasing in t_1, for $t_1 > 0$, implying ρ arbitrarily close to $1 + \sqrt{2}$ if we take n large enough. $\square$

4

Algorithms for on-line routing problems

4.1 Algorithms for the HOLTSP

4.1.1 A best possible algorithm for the HOLTSP on the real line

In this section we present a best possible algorithm for the HOLTSP on the real line with a competitive ratio of $(9 + \sqrt{17})/8$. The algorithm is called WD (for Waiting Deliberately). WD is described completely by its behaviour at the moment a new request is given. The behaviour is determined only by the two unserved extreme requests, one on the positive halfline (*the rightmost extreme*) and one on the negative halfline (*the leftmost extreme*). All other unserved requests will be served while completing the tour and are therefore ignored. If a new request does not define a new extreme it is accordingly also ignored. We take the point 0 as the origin. If a new extreme is on the same side as the WD-server but closer to 0, then this new extreme will be served while completing the tour and is ignored as well. From now on we use the term extreme shortly for a leftmost or rightmost extreme request that is unserved and not ignored. Notice that any request can become extreme only at the moment it is presented.

First we introduce some notation. At any time t,

$$\begin{aligned} p_t &= \text{the position of the WD server,} \\ x_t &= \text{the leftmost extreme, having abcissa } -x_t, \\ y_t &= \text{the rightmost extreme, having abcissa } y_t, \\ X_t &= \text{the leftmost request ever presented until time t,} \\ Y_t &= \text{the rightmost request ever presented until time t.} \end{aligned}$$

Since we use the Euclidean metric on the real line, $d(v, 0) = |v|$ for any point v. We also define

$$\begin{aligned} r_v &= \text{the last time request } v \text{ is given,} \\ \hat{v} &= \max\{d(v, 0), r_v\}, \\ \rho &= (9 + \sqrt{17})/8. \end{aligned}$$

If at time t there is no leftmost extreme (on the negative halfline), we set $x_t = \hat{x}_t = 0$, and similarly we set $y_t = \hat{y}_t = 0$ if there is no rightmost extreme (on the positive halfline). We denote the completion time of WD by Z^{WD} and that of the optimal solution by Z^*.

For notational convenience we will use x_t here, not only for the distance $d(-x_t, 0)$, but also to indicate the request, which actually is at point $-x_t$.

Before giving the precise description of WD, we explain the underlying ideas. Suppose that at time t, when a new request arrives, the position of the WD-server is to the left of the origin, i.e., $p_t \leq 0$ (the case in which $p_t \geq 0$ is symmetrical). WD has to decide which extreme to serve first. Clearly, serving x_t first gives the shortest possible tour at time t. Suppose for the time being that WD decides to serve x_t first. Let t_0 be the moment WD returns in the origin after having served x_t. There is a risk that at time t_0 a new leftmost extreme request arrives. First, this makes serving x_t before t_0 useless, and, second, WD may be too far away from the new request at t_0. If in the optimal off-line solution x_t is served before y_t, then serving x_t first should be a safe option.

Suppose that in the optimal off-line solution y_t is served before x_t. In this case $Z^* \geq \hat{y}_t + y_t + 2x_t$. We distinguish two situations. In the first one $t_0 \leq \rho(\hat{y}_t + y_t + 2x_t) - 2x_t - 2y_t$. Here t_0 is so low that serving x_t first should be a safe option, even if a new leftmost extreme would be presented at t_0.

The second situation occurs if $t_0 > \rho(\hat{y}_t + y_t + 2x_t) - 2x_t - 2y_t$. At t_0 a leftmost extreme at point $-t_0 + \hat{y}_t + y_t$ is given, which the off-line server may reach at t_0 after having served y_t, making $Z^* = t_0 + t_0 - \hat{y}_t - y_t = 2t_0 - \hat{y}_t - y_t$. WD still has to serve both extremes at time t_0, whence $Z^{WD} = t_0 + 2(t_0 - \hat{y}_t - y_t) + 2y_t = 3t_0 - 2\hat{y}_t$. Therefore, for WD to be ρ-competitive,

$$\frac{3t_0 - 2\hat{y}_t}{2t_0 - \hat{y}_t - y} \leq \rho \Leftrightarrow t_0 \geq \frac{\rho y_t - (2 - \rho)\hat{y}_t}{2\rho - 3}. \tag{4.1}$$

This inequality shows the necessity to wait in some cases.

Now suppose WD waits and returns to the origin at $t_0 = \frac{\rho y_t - (2-\rho)\hat{y}_t}{2\rho - 3}$. If no more requests are given, $Z^{WD} = \frac{\rho y_t - (2-\rho)\hat{y}_t}{2\rho-3} + 2y_t$. Since $Z^* \geq \hat{y}_t + y_t + 2x_t$, WD is ρ-competitive if

$$\begin{aligned}\frac{\rho y_t - (2-\rho)\hat{y}_t}{2\rho - 3} + 2y_t \leq \rho(\hat{y}_t + y_t + 2x_t) \;\Leftrightarrow \\ (8\rho - 2\rho^2 - 6)y_t \leq (2\rho^2 - 4\rho + 2)\hat{y}_t + (4\rho^2 - 6\rho)x_t \;\Leftrightarrow \\ (6\rho - 2\rho^2 - 4)y_t \leq (2\rho^2 - 3\rho)x_t.\end{aligned} \tag{4.2}$$

Since, by the choice of ρ, $6\rho - 2\rho^2 - 4 = 2\rho^2 - 3\rho$, inequality (4.2) holds if $x_t \geq y_t$. However, (4.2) does not hold if $x_t < y_t$. We notice that inequality (4.1) is based on the situation in which a new extreme would be presented at t_0. Inequality (4.2) is based on the situation in which at time t the last request was presented. Therefore, (4.2) must be satisfied if WD starts the shortest possible tour at time t first visiting x_t.

Basically, WD tries to satisfy both (4.1) and (4.2). Therefore, in view of (4.2), WD tries to follow the tour that visits the greater extreme first, starting in the origin at a moment such that it remains ρ-competitive and (4.1) and (4.2) are satisfied.

Inequality (4.1) shows that t_0 and therefore the specific moment to leave the origin depends on y_t and $\hat{y}_t$ only. However, to make the analysis of WD easier, we choose the specific moment to leave the origin to depend on x_t, $\hat{x}_t$, y_t and $\hat{y}_t$.

We come to the point now to be more precise about WD. We define

$$\begin{aligned} L_t^- &= \rho\hat{x}_t + (\rho - 2)x_t + (2\rho - 2)y_t, \\ L_t^+ &= \rho\hat{y}_t + (\rho - 2)y_t + (2\rho - 2)x_t. \end{aligned}$$

We notice that

$$x_t \geq y_t \Rightarrow \min\{L_t^-, L_t^+\} + 2x_t \geq (4\rho - 2)y_t = \frac{(2\rho - 2)y_t}{2\rho - 3} \geq \frac{\rho y_t - (2-\rho)\hat{y}_t}{2\rho - 3}. \tag{4.3}$$

Thus, inequality (4.1) is satisfied if WD first serves x_t on a tour that leaves the origin not before time $\min\{L_t^-, L_t^+\}$. (The case $y_t \geq x_t$ is symmetrical.)

We distinguish two basic cases that may occur at time t: $L_t^- \leq L_t^+$ and $L_t^+ \leq L_t^-$ (breaking ties arbitrarily). Each basic case has seven different sub-cases making a total of fourteen cases. Given a basic case, the seven sub-cases form an ordered list. WD acts according to the first case in the list that fits its situation. We give the description of WD by listing the cases and the corresponding actions in Figure 4.1.

The tour that leaves the origin at time $\min\{L_t^-, L_t^+\}$ in the direction of the greater extreme, serves the extreme requests uninterruptedly at maximum speed, and returns to the origin is called the *preferred tour.* The situation in which WD can recover the preferred tour corresponds to cases **I1, I5, II1**, and **II5**.

In cases in which a preferred tour cannot be recovered WD will start an *enforced tour* starting at t in p_t, visiting the extremes uninterruptedly at maximum speed, and returning to the origin. If at time t WD is on the same side as the greater extreme, then WD starts an enforced tour first serving this greater extreme. This tour is the shortest possible tour and therefore inequality (4.2) should be satisfied. Inequality (4.1) is satisfied because WD cannot recover the preferred tour. This situation corresponds to cases **I2, I7, II2**, and **II7**.

If at time t WD is on the same side as the smaller extreme, then WD starts an enforced tour first serving this smaller extreme if certain requirements are met. This situation corresponds to cases **I3, I6, II3**, and **II6**. If these requirements are not met, then WD will cross the origin to serve the greater extreme first. This situation corresponds to cases **I4**, and **II4**.

Before we show that WD is best possible, we state two preliminary lemmas.

Lemma 4.1. *If $L_t^- \leq L_t^+$, $y_t > 0$, and x_t is released at t then case* **I1** *or* **I5** *occurs. If $L_t^+ \leq L_t^-$, $|x_t| > 0$, and y_t is released at t then case* **II1** *or* **II5** *occurs.*

Proof. We give the proof of the first statement only (the proof of the second statement is symmetric). If $x_t \geq y_t$, then $t+d(p_t,x_t) \leq \rho\hat{x}_t+(\rho-2)x_t+(2\rho-2)y_t+x_t$, since $d(p_t,x_t) \leq x_t+y_t$ and $t \leq \hat{x}_t$.

If $x_t < y_t$, then $t+d(p_t,y_t) \leq \rho\hat{x}_t+(\rho-2)x_t+(2\rho-2)y_t+y_t$, since $d(p_t,y_t) \leq x_t+y_t \leq 2y_t$ and using again $t \leq \hat{x}_t$. □

Lemma 4.2. *If $L_t^- \leq L_t^+$, then $Z^* \geq \hat{x}_t+x_t+2y_t$. If $L_t^+ \leq L_t^-$, then $Z^* \geq \hat{y}_t+y_t+2x_t$.*

Proof. The lemma follows directly from the fact that a request can be served neither before its release time nor before its distance to the origin, together with the definitions of L_t^- and L_t^+. □

Theorem 4.3. *WD is ρ-competitive, with $\rho = (9+\sqrt{17})/8$.*

Proof. We prove the theorem by showing that, if WD is ρ-competitive before a new request is given at time t (which is true for $t = 0$), then WD is ρ-competitive after this new request. This is trivially true if the new request is an ignored request. Thus, we only have to be concerned if the new request is either a leftmost or rightmost unserved extreme. Without loss of generality we assume that the new request at time t is rigthmost extreme y_t. Trivial lower bounds on the optimal solution value are then $Z^* \geq t+y_t$ and $Z^* \geq 2X_t+2Y_t$.

Clearly, WD is ρ-competitive if it can recover a preferred tour at time t (cases **I1**, **I5**, **II1** or **II5**). We disregard this situation from now on. If $x_t = 0$, then $Z^{WD} = t+d(p_t,y_t)+y_t \leq 3/2Z^*$, since $Z^* \geq t+y_t$ and $Z^*/2 \geq X_t+Y_t \geq d(p_t,y_t)$. If $|x_t| > 0$, Case **II** at t is dismissed through Lemma 4.1. For the remaining cases, all having $L_t^- \leq L_t^+$ (Case **I**), we have to take the behaviour of the WD-server before t into account. The following claims for the time-interval $[r_{x_t}, t]$ are proved in appendix A.

Case I $L_t^- \leq L_t^+$

I1 $x_t \geq y_t$ and $t + d(p_t, x_t) \leq L_t^- + x_t$. Go in the direction of the origin (or wait in the origin) until being on the preferred tour. At that moment start to follow the preferred tour first serving x_t.

I2 $x_t \geq y_t$ and $p_t \leq 0$. Follow the enforced tour first serving x_t.

I3 $x_t \geq y_t$, $t + 2y_t - p_t \geq (4\rho - 2)x_t$ and $p_t > 0$. Follow the enforced tour first serving y_t.

I4 $x_t \geq y_t$ and $p_t > 0$. Follow the enforced tour first serving x_t.

I5 $y_t > x_t$ and $t + d(p_t, y_t) \leq L_t^- + y_t$. Go in the direction of the origin (or wait in the origin) until being on the preferred tour. At that moment start to follow the preferred tour first serving y_t.

I6 $y_t > x_t$, and $p_t < 0$. Follow the enforced tour first serving x_t.

I7 $y_t > x_t$, and $p_t \geq 0$. Follow the enforced tour first serving y_t.

Case II $L_t^+ \leq L_t^-$ is symmetrical.

II1 $y_t \geq x_t$ and $t + d(p_t, y_t) \leq L_t^+ + y_t$. Go in the direction of the origin (or wait in the origin) until being on the preferred tour. At that moment start to follow the preferred tour first serving y_t.

II2 $y_t \geq x_t$ and $p_t \geq 0$. Follow the enforced tour first serving y_t.

II3 $y_t \geq x_t$, $t + 2x_t - |p_t| \geq (4\rho - 2)y_t$ and $p_t < 0$. Follow the enforced tour first serving x_t.

II4 $y_t \geq x_t$ and $p_t < 0$. Follow the enforced tour first serving y_t.

II5 $x_t > y_t$ and $t + d(p_t, x_t) \leq L_t^+ + x_t$. Go in the direction of the origin (or wait in the origin) until being on the preferred tour. At that moment start to follow the preferred tour first serving x_t.

II6 $x_t > y_t$, and $p_t > 0$. Follow the enforced tour first serving y_t.

II7 $x_t > y_t$, and $p_t \leq 0$. Follow the enforced tour first serving x_t.

Figure 4.1: WD

Claim 4.1. *If there is a $t' \in (r_{x_t}, t]$, at which WD serves a rightmost extreme, then WD is ρ-competitive.*

Claim 4.2. *If there is a $t' \in [r_{x_t}, t]$, at which WD can recover the preferred tour, then WD is ρ-competitive.*

From now on we denote r_{x_t} by τ. Since x_t is still the leftmost unserved request at time t, no new leftmost request is given during the time interval $[\tau, t]$, $x_\tau = x_t$ and $p_\tau > -x_t$. At τ there may be a rightmost unserved extreme y_τ. We may assume that none of the premises of Lemma 4.1 and Claims 4.1 and 4.2 occurs during $[\tau, t]$, since this would make WD directly ρ-competitive. In particular, case **II** occurs at time τ, case **II** does not occur during $(\tau, t]$, cases **I1** and **I5** do not occur during $[\tau, t]$, and WD starts an enforced tour at time τ. We distinguish four main situations.

- *WD starts an enforced tour in the direction of x_t, not turning around before reaching x_t.*
  Thus, $Z^{WD} \leq \tau + d(p_\tau, x_t) + x_t + 2y_t$ and, using Lemma 4.2,
  $$\frac{Z^{WD}}{Z^*} \leq \frac{\tau + x_t + 2y_t}{Z^*} + \frac{d(p_\tau, x_t)}{Z^*} \leq 1 + \frac{X_t + Y_t}{Z^*} \leq \frac{3}{2}.$$

- *WD starts an enforced tour in the direction of x_t, turning around before reaching x_t.*
  To make WD turning around a new rightmost request y_1 must be given at some time $t' \in [\tau, t]$. WD starts to follow an enforced tour at time t' in the direction of y_1. Therefore, case **I3** or **I7** occurs at time t'. In both cases $p_{t'} \geq 0$ by definition.

  The first possibility is that the WD-server does not turn around before he reaches a rightmost extreme. Since we excluded that WD reaches a rightmost extreme before time t, y_t must be given before this rightmost extreme is reached. We note that $p_\tau > p_{t'} \geq 0$, so $Z^{WD} \leq \tau + 2y_t + 2x_t + p_\tau \leq 3/2Z^*$ because $\tau + x_t + 2y_t \leq Z^*$ and $x_t + p_\tau \leq X_t + Y_t \leq Z^*/2$.

  The second possibility is that WD does turn around before reaching a rightmost extreme caused by the release of a new rightmost request y_2 at some time $t'' \in [t', t]$ at which WD sets out on an enforced tour in the direction of x_t. This excludes immediately cases **I3** and **I7** at t'', whereas $p_{t''} > p_{t'} \geq 0$ excludes cases **I2** and **I6**. If at time t' the situation was **I7**, then $y_2 > y_1 > x_t$ excludes case **I4** at t''. If at time t' the situation was **I3**, then $t''+2y_2-p_{t''} > t'+2y_1-p_{t'}$ excludes case **I4** at t''. Thus, this possibility is excluded since we already assumed that cases **I1**,**I5** and **II** do not occur at presenting a new rightmost request in the interval $[\tau, t]$.

- *WD starts an enforced tour in the direction of y_τ, not turning around before reaching a rightmost extreme.*

Since we excluded the premise of Claim 4.1, y_t is given before WD served a rightmost extreme and y_t must be this rightmost extreme. In all cases, since WD remains on enforced tours, $Z^{WD} \leq \tau + 2y_t + |p_\tau| + 2x_t \leq Z^* + x_t + |p_\tau|$, applying Lemma 4.2. If $|p_\tau| \leq y_t$ then $x_t + |p_\tau| \leq \frac{1}{2}Z^*$ and hence $Z^{WD} \leq \frac{3}{2}Z^*$. This is directly true if $p_\tau \geq 0$. If $p_\tau < 0$, the only possible case at time τ in which WD starts an enforced tour in the direction of y_τ is **II4**, which by definition has $y_\tau \geq x_t \geq |p_\tau|$.

- *WD starts an enforced tour in the direction of y_τ, turning around before reaching a rightmost extreme.*
  Since we excluded the premise of Claim 4.1, a new rightmost extreme y_1 must be given at some time $t' \in [\tau, t]$ before WD reaches y_τ. If $y_1 > x_t$, then we have

$$\begin{aligned} L^-_{t'} + y_1 &\geq \hat{x}_t + (2\rho - 3)(x_t + y_1) + 2y_1 \\ &> \tau + (2\rho - 3)(x_t + y_1) + x_t + y_1 \\ &> \tau + d(p_\tau, y_1) = t' + d(p_{t'}, y_1). \end{aligned}$$

  Thus, WD can recover the preferred tour and, using Claim 4.2, is ρ-competitive.

  If $y_1 \leq x_t$, then $y_\tau < y_1 \leq x_t$. This implies that case **II6** is the only possible case at τ, so $p_\tau > 0$ by definition. At t' WD can turn around and reach the origin before time $\tau + 2y_\tau - p_\tau$. At time τ case **II5** did not occur , so by definition $\tau + p_\tau > (2\rho - 2)(y_\tau + x_t)$. Using $4\rho^2 - 5\rho - 2 \geq 0$, we have

$$\begin{aligned} L^-_{t'} &= \hat{x}_t + (\rho - 1)\hat{x}_t + (\rho - 2)x_t + (2\rho - 2)y_1 \\ &> \hat{x}_t + (\rho - 1)[(2\rho - 2)(y_\tau + x_t) - p_\tau] + (\rho - 2)x_t + (2\rho - 2)y_\tau \\ &= \hat{x}_t + 2y_\tau - (\rho - 1)p_\tau + (2\rho^2 - 2\rho - 2)y_\tau + (2\rho^2 - 3\rho)x_t \\ &> \hat{x}_t + 2y_\tau - (\rho - 1)p_\tau + (4\rho^2 - 5\rho - 2)y_\tau > \tau + 2y_\tau - p_\tau. \end{aligned}$$

  Thus, WD can recover the preferred tour and, using Claim 4.2, is ρ-competitive.

□

4.1.2 A best possible algorithm for the HOLTSP on the real line against a fair adversary

We call the best possible algorithm for the HOLTSP against a fair adversary WF (for Waiting under Fairness). WF is the same as WD in Section 4.1.1 only replacing ρ by $\mu = (5 + \sqrt{57})/8$.

Theorem 4.4. *WF is μ-competitive, with $\mu = (5 + \sqrt{57})/8$.*

The proof of μ-competitiveness of WF is exactly the same as the proof of ρ-competitiveness of WD except for the first part of the proof of Claim 4.1. In this part of the proof we have to use the fact that the adversary does not move outside the interval between the leftmost and rightmost request presented in the past. We prove this part of Claim 4.1 for WF in appendix B.

4.1.3 Appendix A

In this appendix we shall prove Claims 4.1 and 4.2. We need a part of Claim 4.1 to prove Claim 4.2 and we need Claim 4.2 to prove the remaining part of Claim 4.1.

Claim 4.1a: Suppose there is a $t' \in (r_{x_t}, t]$, at which WD serves a rightmost extreme and that the last case that occurred before time t' was not case **II6**. Then WD is ρ-competitive.

PROOF. We focus on the earliest possible moment WD can return to the origin after having served a rightmost extreme at time t', which we denote by $t_0^{y'}$. We abuse notation and denote the rightmost extreme served at t' by $y_{t'}$.

The last case that occurred before t' must have been case **I3, I5, I7, II1, II2, II4** or **II6**, since WD moved to the right. The definition of case **I3** implies that $t_0^{y'} \geq (4\rho - 2)x_t$. We have excluded occurrence of case **II6**. In all remaining cases $y_{t'} \geq x_t$, implying (see (4.3), page 35), $\min\{L_{t'}^-, L_{t'}^+\} + 2y_{t'} \geq (4\rho - 2)x_t$, and therefore, as in case **I3**, $t_0^{y'} \geq (4\rho - 2)x_t$. A trivial upper bound on $t_0^{y'}$ is $t + |p_t|$. Thus, we always have

$$(4\rho - 2)x_t \leq t_0^{y'} \leq t + |p_t|. \tag{4.4}$$

Now we consider the situation at time t. Case **I4** at t is excluded since $t + 2y_t - |p_t| > t + |p_t| \geq t_0^{y'} \geq (4\rho - 2)x_t$, implying that case **I3** would have occurred. Thus, WD does not cross the origin after time t before having served any of x_t or y_t. Therefore $Z^{WD} = t + 2x_t + 2y_t - |p_t|$ (unless the preferred tour can be recovered, in which case WD is ρ-competitive). Since $t + y_t \leq Z^*$, it suffices to prove that $2x_t + y_t - |p_t| \leq (\rho - 1)Z^*$.

Suppose first that $y_t \leq \frac{(4-2\rho)x_t}{(2\rho-3)} + |p_t|$. Using $(4\rho - 2) = (2\rho - 2)/(2\rho - 3)$ and

(4.4), we have

$$\begin{aligned}
2x_t + y_t - |p_t| &\leq (2-\rho)(\frac{(4-2\rho)}{(2\rho-3)}x_t + |p_t| - y_t) + 2x_t + y_t - |p_t| \\
&= \frac{(2-\rho)(4-2\rho)}{(2\rho-3)}x_t + (2-\rho)|p_t| - (2-\rho)y_t + 2x_t + y_t - |p_t| \\
&= (\frac{(2\rho^2-4\rho+2)}{(2\rho-3)} - 2)x_t + (2-\rho)|p_t| + (\rho-2)y_t + 2x_t + y_t - |p_t| \\
&= (\rho-1)(\frac{(2\rho-2)}{(2\rho-3)}x_t - |p_t| + y_t) \\
&\leq (\rho-1)(t+y_t) \\
&\leq (\rho-1)Z^*.
\end{aligned}$$

Now suppose $y_t > \frac{(4-2\rho)x_t}{(2\rho-3)} + |p_t|$. Then

$$\begin{aligned}
2x_t + y_t - |p_t| &= (4-2\rho)x_t - (2\rho-3)y_t - |p_t| + (\rho-1)(2x_t+2y_t) \\
&< (4-2\rho)x_t - (4-2\rho)x_t - (2\rho-3)|p_t| - |p_t| + (\rho-1)(2x_t+2y_t) \\
&\leq (\rho-1)(2x_t+2y_t) \\
&\leq (\rho-1)Z^*.
\end{aligned}$$

□

Claim 4.2: If there is a $t' \in [r_{x_t}, t]$, at which WD can recover the preferred tour, then WD is ρ-competitive.

Proof. We denote the *last* moment WD can recover the preferred tour by t'. If there is a rightmost extreme at t' we denote it by y_0. Obviously, WD is ρ-competitive if no new rightmost extremes are given after t', or if WD served a rightmost extreme before t. This is true by Claim 4.1a, since case **II6** occurs only at time τ (Lemma 4.1).

Thus, suppose at $t'' > t'$ a new rightmost extreme y_1 is given before WD reaches an extreme, which causes the WD-server to follow an enforced tour. This excludes case **I1, I5** and **II** by Lemma 4.1. Clearly, $y_1 > y_0$.

Notice that $L^-_{t''} = L^-_{t'} + (2\rho-2)(y_1-y_0)$. Thus, if $L^+_{t'} \leq L^-_{t'}$, then $L^-_{t''} \geq (2\rho-2)(y_1-y_0) + L^+_{t'}$.

If at t' the WD-server would take action to serve y_0 first, then $t' + d(p_{t'}, y_0) \leq \min\{L^-_{t'}, L^+_{t'}\} + y_0$ and $y_0 \geq x_t$ by definition. Clearly, $y_1 > y_0 \geq x_t$ and $t'' + d(p_{t''}, y_1) = t' + d(p_{t'}, y_0) + (y_1 - y_0) < \min\{L^-_{t'}, L^+_{t'}\} + y_0 + (2\rho-2)(y_1-y_0) < L^-_{t''} + y_1$. Thus, WD can recover the preferred tour, which contradicts the assumption that t' is the last time before t at which a preferred tour can be recovered.

If at t' the WD-server would take action to serve x_t first, then $x_t \geq y_0$ by definition. If $y_1 \leq x_t$ the WD-server can recover the preferred tour because $t'' + d(p_{t''}, x_t) = t' + d(p_{t'}, x_t) \leq \min\{L^-_{t'}, L^+_{t'}\} + x_t < L^-_{t''} + x_t$. Again a contradiction.

If $y_1 > x_t$ we have to look at $p_{t''}$. In case $p_{t''} \geq 0$, then $t'' + p_{t''} = t' + p_{t'} \leq \min\{L^-_{t'}, L^+_{t'}\} < L^-_{t''}$. WD can recover the preferred tour, so a contradiction again.

If $p_{t''} < 0$ case **I7** is excluded by definition, whereas $y_1 > x_t$ excludes cases **I2, I3** and **I4**. Therefore, at t'' the only possible case is case **I6**. In this case WD starts an enforced tour first serving x_t and does not turn around unless the preferred tour can be recovered, which is excluded. At t' the WD-server was on the preferred tour or recovering the preferred tour first serving x_t, therefore $Z^{WD} \leq \min\{L^-_{t'}, L^+_{t'}\} + 2x_t + 2y_t$. Clearly WD is ρ-competitive. □

Claim 4.1b: If there is a $t' \in (r_{x_t}, t]$, at which WD serves a rightmost extreme while the last case was **II6**, then WD is ρ-competitive.

PROOF. Case **II6** can only occur at time τ. We therefore may exclude the premises of Lemma 4.1 and Claim 4.1a or 4.2 in the time interval $[\tau, t]$, since this would make WD directly ρ-competitive. We will argue that then $Z^{WD} \leq \tau + 2y_\tau + 2x_t + 2y_t - p_\tau$. This is immediately clear if between τ and t no new requests are given or only requests that make WD starting an enforced tour first serving x_t.

Suppose WD starts following an enforced tour in the direction of a new rightmost request y_1 at some time $t^1 \in [t', t]$, implying that case **I3** or **I7** must occur at t^1. In both cases $p_{t^1} \geq 0$. We assume that t^1 is the first time after t' at which WD goes in the direction of a rightmost extreme, so WD cannot have been to the left of the origin between t' and t^1.

The first possibility is that the WD-server does not turn around before reaching a rightmost extreme. Since we excluded this to occur before time t, y_t must be given before this rightmost extreme is reached. Therefore $Z^{WD} \leq \tau + 2y_\tau + 2x_t + 2y_t - p_\tau$.

The other possibility is that WD turns around before reaching a rightmost extreme caused by the release of a new rightmost request y_2 at some time $t^2 \in [t^1, t]$. This excludes immediately cases **I3** and **I7** at t^2, while $p_{t^2} > p_{t^1} \geq 0$ excludes cases **I2** and **I6**, leaving **I4** as the only possible one at t^2 (we have already excluded all other cases from the beginning). If at time t' the situation was **I7**, then $y_2 > y_1 > x_t$ excludes case **I4** at t_2. If at time t' the situation was **I3**, then $t_2 + 2y_2 - p_{t^2} > t' + 2y_1 - p_{t'}$ excludes case **I4** at t^2. Thus, this possibility is excluded.

We still have to prove that $Z^{WD} \leq \tau + 2y_\tau + 2x_t + 2y_t - p_\tau \leq \rho Z^*$. Using $0 \leq 4\rho^2 - 5\rho - 2$ and $y_\tau < x_t$, we derive the crucial inequality:

$$\begin{aligned}
2y_\tau + x_t - p_\tau &\leq 2y_\tau + x_t - p_\tau + (2-\rho)p_\tau + (4\rho^2 - 5\rho - 2)y_\tau \\
&< (2\rho^2 - 3\rho + 1)x_t + (2\rho^2 - 2\rho)y_\tau + (1-\rho)p_\tau \\
&= (\rho - 1)((2\rho - 2)(x_t + y_\tau) - p_\tau + x_t + 2y_\tau) \\
&< (\rho - 1)(\hat{x}_t + x_t + 2y_\tau). \qquad (4.5)
\end{aligned}$$

Suppose first that $y_\tau \leq y_t$. Applying this bound in (4.5) and using the fact that $\tau + x_t + 2y_t \leq Z^*$ yields $Z^{WD} \leq \rho Z^*$.

Now suppose that $y_t < y_\tau$. If in the optimal solution y_τ is served after x_t, then $Z^* \geq \hat{x}_t + x_t + 2y_\tau$. We notice that (4.5) also holds if y_τ is substituted by y_t, i.e., $2y_t + x_t - p_\tau \leq (\rho - 1)(\hat{x}_t + x_t + 2y_\tau)$. Therefore, $2y_t + x_t - p_\tau \leq (\rho - 1)Z^*$. These observations together yield $Z^{WD} \leq \rho Z^*$.

If $y_t < y_\tau$ and in the optimal solution y_τ is served before x_t then $Z^* \geq t + y_t + 2x_t$, in case y_t is also served before x_t, or, if not,

$$Z^* \geq \hat{y}_\tau + y_\tau + 2x_t + 2y_t. \tag{4.6}$$

In the former case $Z^{WD} \leq \frac{3}{2}Z^*$, following easily from the observation that $\tau + y_\tau - p_\tau \leq t$. In the latter case we have to take the behavior of WD before τ into account, in particular on the time interval $[r_{y_\tau}, \tau]$. We denote r_{y_τ} by t^3 and the leftmost extreme at time t^3 by x_3. We note that in $(t^3, \tau]$ only new leftmost extremes can be given.

If during $[t^3, \tau]$ WD never moves to the left, then at t^3 WD either starts moving to the right until y_τ is reached or WD waits some time in the origin to recover the preferred tour. Therefore, $Z^{WD} \leq t^3 + d(p_{t^3}, y_\tau) + y_\tau + 2x_t + 2y_t = t^3 + y_\tau + 2x_t + 2y_t + d(p_{t^3}, y_\tau) \leq \frac{3}{2}Z^*$, since $t^3 + y_\tau + 2x_t + 2y_t \leq Z^*$ and $d(p_{t^3}, y_\tau) \leq X_t + Y_t \leq Z^*/2$, or $Z^{WD} \leq \min\{L_\tau^-, L_\tau^+\} + 2y_\tau + 2x_t + 2y_t \leq \rho Z^*$.

If WD does move to the left during $[t^3, \tau]$ we define time $t^l \in [t^3, \tau]$ as the last moment before time τ at which this happens.

First suppose that during the interval $[t^l, \tau]$ WD can recover the preferred tour. Let $t^p \in [t^l, \tau]$ be the last moment at which this is the case. If at t^p WD follows (or recovers) the preferred tour first serving y_τ, then he is still doing so until τ and $Z^{WD} \leq \min\{L_\tau^-, L_\tau^+\} + 2y_\tau + 2x_t + 2y_t \leq \rho Z^*$. If at t^p WD recovers the preferred tour first serving x_{t^p}, then $p_{t^p} \leq 0$ and $x_{t^p} \geq y_\tau$. This excludes $p_\tau > 0$ at time τ, conflicting the premises of the claim we are proving.

Thus, from now on we assume that during $[t^l, \tau]$ the preferred tour cannot be recovered. We consider first the case that $p_{t^l} \geq 0$. Since at t^l WD followed an enforced tour first serving the leftmost extreme, the last case that occurred before t^l must have been case **I4**. In case **I4** the premise is that $L^- \leq L^+$ and therefore Lemma 4.1 implies that the last request before t^l must have been y_τ. Therefore, $Z^{WD} \leq t^3 + p_{t^3} + 2y_\tau + 2x_t + 2y_t \leq 3/2\ Z^*$, since $p_{t^3} < y_\tau < x_t$.

Now consider the case that $p_{t^l} < 0$. We distinguish three situations.

- *WD serves x_{t^l} before time τ, while the last case was not case* **I6**.
  The symmetry of WD allows to use the same analysis used for Claim 4.1a to prove ρ-competitiveness, by substituting y_τ for x_t and x_t for y_t.

- *WD serves x_{t^l} before time τ or WD turns around before reaching x_{t^l} , while*

case **I6** *is the last case before t^l.*
Using the same arguments as before Lemma 4.1 implies that the last request must have been y_τ. By definition of case **I6**, $p_{t^3} < 0$, and hence $Z^{WD} \leq t^3 + 2x_{t^l} + 2y_\tau + 2x_t + 2y_t - |p_{t^3}|$. Since, $Z^* \geq \hat{y}_\tau + y_\tau + 2x_t + 2y_t$, we are left to prove that $2x_{t^l} + y_\tau - |p_{t^3}| \leq (\rho - 1)Z^*$. At t^3 case **I5** did not occur, so by definition $t^3 + |p_{t^3}| > (2\rho - 2)(x_{t^l} + y_\tau)$. Using $x_t > y_\tau > x_{t^l}$ and $0 \leq 4\rho^2 - 5\rho - 2$, we have

$$\begin{aligned} 2x_{t^l} + y_\tau - |p_{t^3}| &\leq 2x_{t^l} + y_\tau - (\rho - 1)|p_{t^3}| + (4\rho^2 - 5\rho - 2)x_{t^l} \\ &< (2\rho^2 - 4\rho + 2)(x_{t^l} + y_\tau) + (\rho - 1)(y_\tau + 2x_t - |p_{t^3}|) \\ &= (\rho - 1)((2\rho - 2)(x_{t^l} + y_\tau) - |p_{t^3}| + y_\tau + 2x_t) \\ &< (\rho - 1)(t^3 + y_\tau + 2x_t + 2y_t) \leq (\rho - 1)Z^*. \end{aligned}$$

- *WD turns around before reaching a leftmost extreme.*
At some time $t^4 \in (t^l, \tau]$ a new leftmost request x_4 must be given such that WD starts an enforced tour in the direction of y_τ. This excludes cases **I**, **II1**, **II3**, **II5**, and **II7**. Case **II2** and **II6** at t^4 are excluded, since $p_{t^4} < 0$. If at t^4 case **II4** occurs, then by definition $y_\tau \geq x_4 > x_{t^l}$. This immediately excludes cases **I1**, **I2**, **I3**, **I4**, **II5**, **II6**, and **II7** as the last case before t^4. At t^l WD is going in the direction of the leftmost extreme. This excludes case **I5** **I7**, **II1**, **II2**, and **II4** as the last case before t^4. We already proved ρ-competitiveness if the last case before t^4 is case **I6**. If the last case before t^4 is case **II3**, then at t^4 case **II3** occurs instead of case **II4** since $t^4 + 2x_4 - |p_{t^4}| > t^l + 2x_{t^l} - |p_{t^l}|$. □

4.1.4 Appendix B

In this appendix we shall prove the first part of Claim 4.1 for WF. We use this proof in the same way we use Claim 4.1a for WD.

Claim 4.1a: Suppose there is a $t' \in (r_{x_t}, t]$, at which WF serves a rightmost extreme and that the last case that occurred before time t' was not case **II6**. Then WF is μ-competitive.

Proof. We denote the time before t at which WF serves a rightmost extreme by t'. We abuse notation and denote this rightmost extreme by $y_{t'}$. We denote $t_0^{y'}$ as the earliest possible moment WF can return to the origin after having served $y_{t'}$.

If at t' WF serves a rightmost extreme the last case that occurred before t' must have been case **I3, I5, I7, II1, II2, II4** or **II6**. If case **I3** is the last case before t', then by definition

$$t_0^{y'} \geq (4\mu - 2)x_t. \tag{4.7}$$

We have excluded occurrence of case **II6**. Therefore, if case **I3** is not the last case before t', then $y_{t'} \geq x_t$ and

$$t_0^{y'} \geq \min\{L_{t'}^-, L_{t'}^+\} + 2y_{t'}. \tag{4.8}$$

We note that, if $y_{t'} \geq x_t$, then $\min\{L_{t'}^-, L_{t'}^+\} + 2y_{t'} \geq (2\mu - 2)(x_t + y_{t'}) + 2y_{t'} \geq (4\mu - 2)x_t$. Thus, we always have $t_0^{y'} \geq (4\mu - 2)x_t$. This excludes occurrence of case **I4** after t'.

We now assume that t' is the *last* time before t at which WF serves a rightmost extreme while the last case that occurred before time t' was not case **II6**.

If WF can recover the preferred tour during the time interval $[t', t]$, then WF is μ-competitive. This follows from the proof of Claim 4.2, since in $[t', t]$ no extreme is served. We therefore assume WF cannot recover the preferred tour after t' for the remainder of the proof.

We denote the first request after t' by y^n. If at r_{y^n} WF starts an enforced tour first serving x_t, then case **I2** or case **I6** must occur. In both cases WF does not turn around before t. If at r_{y^n} WF starts an enforced tour first serving y^n, then case **I3** or case **I7** must occur. Also in these cases WF does not turn around before t (cf. proof of Claim 4.2). Thus, we have

$$Z^{WF} = r_{y^n} + 2x_t + 2y_t - |p_{t_{y^n}}| = t + 2x_t + 2y_t - |p_t|. \tag{4.9}$$

Since $t_0^{y'} - |p_t| + y_t \leq t + y_t \leq Z^*$, it suffices to prove that $2x_t + y_t \leq (\mu - 1)(t_0^{y'} + y_t)$, or equivalently, $t_0^{y'} \geq \frac{2x_t + (2-\mu)y_t}{(\mu-1)}$.

If $y_t \leq x_t$ we use (4.7) or (4.8) to obtain

$$t_0^{y'} \geq (4\mu - 2)x_t \geq \frac{2x_t + (2-\mu)y_t}{(\mu - 1)}.$$

If $y_t > x_t$ and $y_{t'} \geq y_t$, then case **I3** is excluded as the last case before t', since $y_{t'} > x_t$. We use (4.8) to obtain

$$t_0^{y'} \geq (2\mu - 2)(x_t + y_{t'}) + 2y_{t'} \geq \frac{2x_t + (2-\mu)y_t}{(\mu - 1)}.$$

If $y_t > x_t$ and $y_{t'} < y_t$, then we have to use the fact that the adversary is fair. We denote $t_0^{Y'}$ as the earliest possible moment WF can return to the origin after having served $Y_{t'}$. We distinguish three situations.

- $y_{t'} = Y_{t'}$.
  At r_{y^n} the position of the optimal server cannot be to the right of $Y_{t'}$. Therefore $Z^* \geq r_{y^n} + 2y_t - y_{t'}$ and inserting that into (4.9) shows that it suffices to prove that $2x_t + y_{t'} - |p_{t_{y^n}}| \leq (\mu - 1)Z^*$ or, using $t_0^{y'} - |p_{t_{y^n}}| + y_t \leq t + y_t \leq Z^*$, to prove that $t_0^{y'} \geq \frac{2x_t + (2-\mu)y_{t'}}{(\mu-1)}$.

If case **I3** is the last case before t', then $y_{t'} \leq x_t$. We use (4.7) to obtain

$$t_0^{y'} \geq (4\mu - 2)x_t \geq \frac{2x_t + (2-\mu)y_{t'}}{(\mu - 1)}.$$

If case **I3** is not the last case before t', then $y_{t'} \geq x_t$ and, using (4.8), we obtain

$$t_0^{y'} \geq (2\mu - 2)(x_t + y_{t'}) + 2y_{t'} \geq \frac{2x_t + (2-\mu)y_{t'}}{(\mu - 1)}.$$

- $y_{t'} < Y_{t'}$ and $y_t \leq Y_{t'}$.
We denote the time at which WF serves $Y_{t'}$ for the last time by t''. If $t'' > \tau$ then the last case before t'' cannot be case **I3**, since this would require $y_t > x_t \geq Y_{t'}$. We excluded case **II6** as the last case before serving a rightmost extreme after τ, so $Y_{t'} \geq x_t$ in all other cases. We use (4.8) to obtain

$$t_0^{y'} > t_0^{Y'} \geq (2\mu - 2)(x_t + Y_{t'}) + 2Y_{t'} \geq \frac{2x_t + (2-\mu)y_t}{(\mu - 1)}.$$

If $t'' \leq \tau$, then we focus on the last case before t''. If this is case **II6**, then $x_{t''} > Y_{t'}$ by definition. Since $x_{t''} > Y_{t'} \geq y_t > x_t$, WF must have served $x_{t''}$ before τ, therefore $\tau \geq 2Y_{t'} + x_{t''}$. We have

$$\begin{aligned} 2x_t + y_t &< (\mu - 1)(4x_t + 2y_t) < (\mu - 1)(2Y_{t'} + x_{t''} + x_t + 2y_t) \\ &\leq (\mu - 1)(\tau + x_t + 2y_t) \leq (\mu - 1)Z^*. \end{aligned}$$

If the last case before t'' is not case **II6**, then $t_0^{Y'} \geq 2\mu Y_{t'}$ by definition. This can easily be verified by checking all cases. Since x_t and $y_{t'}$ are given after t'' we have

$$\tau \geq 2\mu Y_{t'} - p_\tau, \tag{4.10}$$

and

$$r_{y_{t'}} \geq 2\mu Y_{t'} - p_\tau. \tag{4.11}$$

We focus on the last case before t'. If case **I3** is not the last case before t', then $y_{t'} \geq x_t$. Using (4.8), and (4.10) or (4.11) we obtain

$$t_0^{y'} \geq (2\mu - 2)(2\mu - 1)Y_{t'} + 2\mu y_{t'} \geq \frac{2x_t + (2-\mu)y_t}{(\mu - 1)}.$$

If case **I3** is the last case before t', then we look at the time interval $[\tau, t']$. Suppose WF can recover the preferred tour during the interval $[\tau, t']$. Let $t^p \in [\tau, t']$ be the last time at which this is the case. In the proof of Claim 4.2 we have seen that, if WF can recover the preferred tour first serving x_t, then

WF does not turn around unless a preferred tour can be recovered. If WF can recover the preferred tour first serving y_p, then $t_0^{y'} > t_0^{y^p} \geq \min\{L^-_{t^p}, L^+_{t^p}\}+2y_{t^p}$ and $y_p \geq x_t$. Since y_p is given after t'' we have $t_{y_p} \geq (2\mu-1)Y_{t'}$. Using the same analysis as for the situation that case **I3** is not the last case before t' we can show μ-competitiveness for WF. We therefore assume there is no $t^p \in [\tau, t']$.

Thus, suppose first that at τ WF starts an enforced tour first serving x_t. Clearly, WF must turn around before reaching x_t, so a new rightmost extreme y^q at some time $t_q \in [\tau, t']$ must be given such that WF starts an enforced tour first serving y^q, implying case **I3** or **I7** occurs at t_q. If at t_q case **I3** occurs, then p_{t_q} and $p_\tau > 0$. Case **I1** did not occur and $Y_{t'} \geq p_\tau$, so $\tau + Y_{t'} \geq \tau + p_\tau = t_q + p_{t_q} > \mu\hat{x}_t + (\mu-2)x_t + (2\mu-2)y^q$ by definition. If we combine this with (4.10), we obtain

$$y^q < \frac{(3\mu-2\mu^2)Y_{t'} + (2-\mu)x_t}{2\mu-2}. \tag{4.12}$$

By definition of case **I3** $\tau + Y_{t'} + 2y^q > t_q + 2y^q - p_{t_q} \geq (4\mu-2)x_t$ and using (4.12) we have

$$\tau \quad > \quad (4\mu-2)x_t - Y_{t'} - 2\frac{(3\mu-2\mu^2)Y_{t'} + (2-\mu)x_t}{2\mu-2}. \tag{4.13}$$

Suppose first $Y_{t'} \leq 2x_t$. We use $4\mu^2 - 5\mu - 2 = 0$ and (4.13) to obtain

$$\begin{aligned}
2x_t + y_t &= 2x_t + y_t + (4\mu^2-5\mu-2)Y_{t'} \\
&\leq 2x_t + y_t + (4\mu^2-2\mu-6)x_t + (2\mu^2-4\mu+1)Y_{t'} \\
&< (\mu-1)[(4\mu-2)x_t - Y_{t'} - 2[\frac{(3\mu-2\mu^2)Y_{t'} + (2-\mu)x_t}{2\mu-2}] + x_t + 2y_t] \\
&\leq (\mu-1)(\tau + x_t + 2y_t) \leq (\mu-1)Z^*.
\end{aligned}$$

If $Y_{t'} > 2x_t$, we use (4.10) to obtain

$$\begin{aligned}
2x_t + y_t &< (4\mu^2-5\mu+1)x_t + (\mu-1)2y_t \\
&= (\mu-1)((2\mu-1)2x_t + x_t + 2y_t) \\
&< (\mu-1)((2\mu-1)Y_{t'} + x_t + 2y_t) \\
&\leq (\mu-1)(\tau + x_t + 2y_t) \\
&\leq (\mu-1)Z^*.
\end{aligned}$$

If at t_q case **I7** occurs, then $y^q > x_t$ and WF cannot recover the preferred tour. Therefore, using (4.10), we have

$$\begin{aligned}
t_0^{y'} &> L^-_{t^q} + 2\max\{y_{t'}, y^q\} > (2\mu-2)(2\mu-1)Y_{t'} + 2\max\{y_{t'}, y^q\} \\
&> \frac{2x_t + (2-\mu)y_t}{(\mu-1)}.
\end{aligned} \tag{4.14}$$

If at τ WF starts an enforced tour first serving y_τ, then, by Lemma 4.1, case **II** occurs. If $y_\tau \geq x_t$, then, similar to (4.14),

$$t_0^{y'} \geq L_\tau^+ + 2\max\{y_{t'}, y_\tau\} \geq \frac{2x_t + (2-\mu)y_t}{(\mu-1)}. \tag{4.15}$$

If $y_\tau < x_t$, then $\tau + p_\tau > \mu\hat{y}_\tau + (\mu-2)y_\tau + (2\mu-2)x_t$ by definition of case **II**, which we assumed not to occur. We use (4.11) to obtain

$$\begin{aligned} \tau &> (2\mu^2-\mu)Y_{t'} + (\mu-2)y_\tau + (2\mu-2)x_t - p_\tau \\ &> (2\mu^2-3)Y_{t'} + (2\mu-2)x_t. \end{aligned} \tag{4.16}$$

Using (4.16), $y_t \leq Y_{t'}$, and $2\mu^3 - 4\mu - 1 > 0$ we obtain

$$\begin{aligned} 2x_t + y_t &< 2x_t + y_t + (2\mu^3 - 4\mu - 1)x_t \\ &< 2x_t + y_t + (2\mu^3 - 2\mu^2 - \mu)y_t + (2\mu^2 - 3\mu - 1)x_t \\ &\leq (2\mu^3 - 2\mu^2 - 3\mu + 3)Y_{t'} + (2\mu^2 - 3\mu + 1)x_t + (2\mu - 2)y_t \\ &= (\mu-1)[(2\mu^2-3)Y_{t'} + (2\mu-2)x_t + x_t + 2y_t] \\ &\leq (\mu-1)(\tau + x_t + 2y_t) \leq (\mu-1)Z^*. \end{aligned} \tag{4.17}$$

- $y_{t'} < Y_{t'}$ and $y_t > Y_{t'}$.
  At t_{y^n} the position of the optimal server cannot be to the right of $Y_{t'}$. Therefore $Z^* \geq t_{y^n} + 2y_t - Y_{t'}$ and it suffices to prove that $2x_t + Y_{t'} - |p_{t_{y^n}}| \leq (\mu-1)Z^*$ or, using $t_0^{y'} - |p_{t_{y^n}}| + y_t \leq t + y_t \leq Z^*$, to prove that $t_0^{y'} \geq \frac{2x_t + (2-\mu)Y_{t'}}{(\mu-1)}$.

  We use the same analysis as for the previous situation ($y_{t'} < Y_{t'}$ and $y_t \leq Y_{t'}$). The only difference is that we now have to prove that $2x_t + Y_{t'} \leq (\mu-1)Z^*$ or $t_0^{y'} \geq \frac{2x_t+(2-\mu)Y_{t'}}{(\mu-1)}$, instead of having to prove that $2x_t + y_t \leq (\mu-1)Z^*$ or $t_0^{y'} \geq \frac{2x_t+(2-\mu)y_t}{(\mu-1)}$. This can be done by substituting $Y_{t'}$ for y_t in the two equations mentioned above, in the proof of the previous situation. In the proof of the previous situation we excluded some cases, using $y_t \leq Y_{t'}$. These cases we treat separately.

  If $t'' > \tau$, and the last case before t'' is case **I3**, then $x_t \geq Y_{t'}$. We use (4.7) to obtain
  $$t_0^{y'} > t_0^{Y'} \geq (4\mu-2)x_t \geq \frac{2x_t + (2-\mu)Y_{t'}}{(\mu-1)}.$$

  If $t'' \leq \tau$ and the last case before t'' is case **II6**, we excluded the possibility that $x_t \geq x_{t''}$. If $x_t \geq x_{t''}$ and case **I3** is the last case before t', then we use

(4.7) to obtain

$$t_0^{y'} \geq (4\mu - 2)x_t \geq \frac{2x_t + (2-\mu)Y_{t'}}{(\mu - 1)}.$$

If case **I3** is not the last case before t', then $y_{t'} \geq x_t$, which contradicts $x_t \geq x_{t''} > Y_{t'}$.

In inequality (4.17) we also used $y_t \leq Y_{t'}$. Using (4.16) and $2\mu^3 - 4\mu - 1 > 0$ we obtain

$$\begin{aligned} 2x_t + Y_{t'} &< 2x_t + Y_{t'} + (2\mu^3 - 4\mu - 1)\min\{x_t, Y_{t'}\} \\ &< (2\mu^3 - 2\mu^2 - 3\mu + 3)Y_{t'} + (2\mu^2 - 3\mu + 1)x_t + (2\mu - 2)y_t \\ &= (\mu - 1)[(2\mu^2 - 3)Y_{t'} + (2\mu - 2)x_t + x_t + 2y_t] \\ &\leq (\mu - 1)(\tau + x_t + 2y_t) \leq (\mu - 1)Z^*. \end{aligned}$$

□

4.2 Algorithms for the NOLTSP

4.2.1 An algorithm for the NOLTSP in general metric spaces

In this section we present a $(\sqrt{2}+1)$-competitive algorithm for the NOLTSP in general metric spaces. The algorithm is called RH (for Return Home). We denote the optimal tour over all request presented up till time t by T_t^*.

Algorithm Return Home

At any moment t at which RH receives a new request, he returns to the origin via the shortest path. Once in the origin at time t_0 he computes the optimal tour $T_{t_0}^*$ over all the requests presented up till time t_0. RH starts to follow this tour $T_{t_0}^*$, staying within distance $(\sqrt{2}-1)t'$ of the origin at any time t', by adjusting his speed at the latest possible time.

Theorem 4.5. *Algorithm Return Home is $(\sqrt{2}+1)$-competitive for the NOLTSP in general metric spaces.*

Proof. Let time t be the time at which the last request is given. Obvious lower bounds on the optimal solution value are $Z^* \geq t$ and $Z^* \geq |T_t^*|$. We distinguish two situations.

- *RH does not have to adjust his speed after time t.*
  Since the distance of RH to the origin at time t is at most $(\sqrt{2}-1)t$, we have $Z^{RH} \leq t + (\sqrt{2}-1)t + |T_t^*| \leq (\sqrt{2}+1)Z^*$.
- *RH has to adjust his speed after time t.*
  Suppose $\sigma_w = (t_w, x_w)$ is the last request causing RH to adjust his speed. RH adjusts his speed at the latest possible time, so he serves σ_w at time $w = (\sqrt{2}+1)x_w$. After time w RH continues to follow T_t^*. Let $T(w)$ denote

the unfinished part of tour T_t^* at time w. We have $Z^{RH} \leq (\sqrt{2}+1)x_w + |T(w)|$, whereas $Z^* \geq x_w + |T(w)|$. Therefore, we have $Z^{RH} \leq (\sqrt{2}+1)Z^*$.

□

4.2.2 An algorithm for the NOLTSP on the real line

We present an algorithm for the NOLTSP on the real line with a competitive ratio of 2.06. The algorithm is called WA (for Wait and Anticipate). The behaviour of WA is determined only by the two unserved extreme requests, one to the right of the WA-server (*the rightmost extreme*) and one to the left of the WA-server (*the leftmost extreme*). All other unserved requests will be served while completing the tour and are therefore ignored. If a new request does not define a new extreme it is accordingly also ignored. We take the point 0 as the origin. From now on we use the term extreme shortly for a leftmost or rightmost extreme request that is unserved and not ignored. Notice that any request can become extreme only at the moment it is presented.

First we introduce some notation. At any time t,

$$\begin{aligned}
p_t &= \text{the position of the WA server,}\\
x_t &= \text{the leftmost extreme,}\\
y_t &= \text{the rightmost extreme,}\\
I_t &= \text{the interval } [x_t, y_t],\\
X_t &= \text{the leftmost request ever presented until time t,}\\
Y_t &= \text{the rightmost request ever presented until time t.}
\end{aligned}$$

Since we use the Euclidean metric on the real line, $d(v,0) = |v|$ for any point v. We also define

$$\begin{aligned}
r_v &= \text{the last time request } v \text{ is given,}\\
\hat{v} &= \max\{d(v,0), r_v\},\\
\rho &= 2.06.
\end{aligned}$$

We denote the completion time of WA by Z^{WA} and that of the optimal solution by Z^*. We notice that

$$\hat{x}_t \leq \hat{y}_t \Rightarrow Z^* \geq \hat{x}_t + d(x_t, y_t). \tag{4.18}$$

For notational convenience we will use x_t here, not only for the distance $d(x_t, 0)$, but also to indicate the request, which actually is at point $+x_t$ or at point $-x_t$. Similarly, we use p_t for the distance $d(p_t, 0)$ and the position of the WA server, which actually is at point $+p_t$ or at point $-p_t$.

WA is based on the same ideas as the lower bound proof. The lower bound proof is basically about an on-line server always serving the wrong extreme first. In order to be ρ-competitive, an on-line server has to anticipate on this. In the lower bound proof we also have seen that situations exist, in which any on-line server has to remain idle while there are still unserved requests, in order to be best possible. Therefore, the WA-server waits as long as possible anticipating on the fact that he serves the wrong extreme first. We define

$$\begin{aligned} L_t^x &= \min\{\rho\hat{x}_t + (\rho-2)x_t + (\rho-1)y_t, \rho\hat{y}_t + (\rho-2)y_t + (\rho-3)x_t\}, \\ L_t^y &= \min\{\rho\hat{y}_t + (\rho-2)y_t + (\rho-1)x_t, \rho\hat{x}_t + (\rho-2)x_t + (\rho-3)y_t\}. \end{aligned}$$

If there is no rightmost extreme we set $L_t^x = (\rho-1)\hat{x}_t$, if there is no leftmost extreme we set $L_t^y = (\rho-1)\hat{y}_t$.

If $\hat{x}_t \leq \hat{y}_t$, then at time $\hat{x}_t + d(x_t, y_t)$ the optimal server can be in point y_t. So, if at time $\hat{x}_t + d(x_t, y_t)$ a new request in point y_t is given, we still have $Z^* \geq \hat{x}_t + d(x_t, y_t)$. The case $\hat{y}_t \leq \hat{x}_t$ is symmetrical.

Suppose $\hat{x}_t \leq \hat{y}_t$, I_t contains 0, and the WA-server leaves the origin at $L_t^x = \rho\hat{x}_t + (\rho-2)x_t + (\rho-1)y_t$ on a tour first serving x_t. The optimal server, having served x_t, can be in point y_t at time $\hat{x}_t + x_t + y_t$. If, at time $\hat{x}_t + x_t + y_t$, a new request in point at distance d_y to the right of y_t is presented, then $Z^* \geq \hat{x}_t + x_t + y_t + d_y$. We have $Z^{WA} \leq \rho\hat{x}_t + (\rho-2)x_t + (\rho-1)y_t + 2x_t + y_t + d_y = \rho(\hat{x}_t + x_t + y_t) + d_y$. Clearly, $Z^{WA}/Z^* \leq \rho$.

If the optimal server serves y_t before x_t, then at time $\hat{y}_t + x_t + y_t$ a new request at distance d_x to the left of point x_t is given, so $Z^* \geq \hat{y}_t + x_t + y_t + d_x$. We use the definition of L_t^x, to obtain $Z^{WA} \leq \rho\hat{y}_t + (\rho-2)y_t + (\rho-3)x_t + 2x_t + y_t + x_t + y_t + d_x = \rho(\hat{y}_t + x_t + y_t) + d_x$. Clearly, $Z^{WA}/Z^* \leq \rho$ also in this case.

If the WA-server leaves the origin at $L_t^x = \rho\hat{y}_t + (\rho-2)y_t + (\rho-3)x_t$ on a tour first serving x_t, then, combined with the definition of L_t^x, $Z^{WA} = \rho\hat{y}_t + (\rho-1)y_t + (\rho-1)x_t \leq \rho(\hat{x}_t + x_t + y_t)$. If the optimal server serves x_t before y_t, then clearly $Z^{WA}/Z^* \leq \rho$. If in the optimal tour y_t is served before x_t, then we can use the same arguments as above to show that $Z^{WA}/Z^* \leq \rho$.

Basically, WA tries to follow the same tour as the optimal off-line server, leaving the origin at the last possible moment such that he is ρ-competitive if no new requests are given and such that he is ρ-competitive if he has to serve the extreme he served first again. (Because it turns out he did not follow the same tour as the optimal off-line server.) L_t^x and L_t^y are chosen such that these requirements are met.

We distinguish three possible situations that may occur at time t: $p_t \notin I_t$, $p_t \in I_t$ but $0 \notin I_t$, and $p_t \in I_t$ and $0 \in I_t$. Each possible situation has some subcases, which form an ordered list. WA acts according to the first case in the list that fits its situation. We give the description of WA by listing the cases and the corresponding actions in Figure 4.2.

If $\hat{x}_t \leq \hat{y}_t$ (the case $\hat{y}_t \leq \hat{x}_t$ is symmetrical) then we call the tour that leaves the origin at time L_t^x in the direction of x_t and serves the extreme requests uninterruptedly at maximum speed the *preferred tour*. We call the tour that leaves the origin at time L_t^y in the direction of y_t and serves the extreme requests uninterruptedly at maximum speed the *anticipating tour*.

In cases in which a preferred tour or an anticipating tour cannot be recovered, WA will start an *enforced tour* starting at t in p_t and visiting the extremes uninterruptedly at maximum speed.

We state two preliminary lemmas.

Lemma 4.6. *If $\hat{x}_t \leq \hat{y}_t$, then $L_t^y = \rho\hat{x}_t + (\rho - 2)x_t + (\rho - 3)y_t$. If $\hat{y}_t \leq \hat{x}_t$, then $L_t^x = \rho\hat{y}_t + (\rho - 2)y_t + (\rho - 3)x_t$.*

PROOF. We give the proof of the first statement only (the proof of the second statement is symmetric). If $\hat{x}_t \leq \hat{y}_t$, then $\rho\hat{y}_t + (\rho - 2)y_t + (\rho - 3)x_t \leq \rho\hat{x}_t + (\rho - 2)x_t + (\rho - 1)y_t$. □

Lemma 4.7. *At any time t, $d(p_t, 0) < 0.54t$.*

PROOF. If the ratio $d(p_t, 0)/t$ reaches a local maximum, then it must be a moment at which WA serves an extreme. Suppose WA serves the rightmost extreme y_t (the case in which WA serves the leftmost extreme x_t is symmetrical). It suffices to prove that WA serves y_t at time $t > y_t/0.54$. The possible last cases before time t are cases **Iy1**, **Iy2**, **IIIx2**, **IIIy1** and **IIIy3**.

If case **Iy1** or **Iy2** is the last case before time t, then WA reaches y_t at $t \geq L_t^y + y_t = (\rho - 1)\hat{y}_t + y_t > y_t/0.54t$. If case **IIIy1** or **IIIy3** is the last case before time t, then WA reaches y_t at $t \geq L_t^y + y_t = \min\{\rho\hat{y}_t + (\rho-2)y_t + (\rho-1)x_t, \rho\hat{x}_t + (\rho-2)x_t + (\rho - 3)y_t\} > y_t/0.54t$. If case **IIIx2** is the last case before time t, then WA reaches y_t at $t \geq L_t^y + y_t = \min\{\rho\hat{y}_t + (\rho-2)y_t + (\rho-1)x_t, \rho\hat{x}_t + (\rho-2)x_t + (\rho-3)y_t\} = \rho\hat{x}_t + (\rho - 2)x_t + (\rho - 2)y_t$. We use $\hat{y}_t \leq 1.17x_t$, to obtain $t > y_t/0.54$. □

Theorem 4.8. *WA is ρ-competitive, with $\rho = 2.06$.*

PROOF. We prove the theorem by showing that, if WA is ρ-competitive before a new request is given at time t (which is true for $t = 0$), then WA is ρ-competitive after this new request. This is trivially true if the new request is an ignored request. Thus, we only have to be concerned if the new request is either a leftmost or rightmost unserved extreme. Without loss of generality we assume that the new request at time t is a rightmost extreme y_t. Trivial lower bounds on the optimal solution value are then $Z^* \geq \hat{y}_t$, $Z^* \geq \min\{X_t, Y_t\} + X_t + Y_t$, and $Z^* \geq \min\{\hat{x}_t, \hat{y}_t\} + d(x_t, y_t)$.

Clearly, WA is ρ-competitive if it can recover a preferred tour or an anticipating tour at time t (cases **Iy1**, **IIIx1**, **IIIx2**, **IIIy1** or **IIIy2**). We disregard this situation from now on. We prove ρ-competitiveness for each remaining case, in the order given by Figure 4.2.

From now on, if there is a leftmost extreme x_t at time t, we denote r_{x_t} by τ. Since x_t is still the leftmost unserved request at time t, no new leftmost request is given during the time interval $[\tau, t]$, $x_\tau = x_t$ and $p_{t'} > x_t$ $\forall t' \in [\tau, t]$. At τ there may be a

Situation I $p_t \notin I_t$

Ix1 x_t is the only extreme and $t + d(p_t, x_t) \leq L_t^x + x_t$. Go in the direction of the origin (or wait in the origin) until being on the preferred tour. At that moment start to follow the preferred tour first serving x_t.

Ix2 x_t is the only extreme. Follow the enforced tour serving x_t.

Iy1 y_t is the only extreme and $t + d(p_t, y_t) \leq L_t^y + y_t$. Go in the direction of the origin (or wait in the origin) until being on the preferred tour. At that moment start to follow the preferred tour first serving y_t.

Iy2 y_t is the only extreme. Follow the enforced tour serving y_t.

Situation II $p_t \in I_t$ **and** $0 \notin I_t$

Serve the extreme which is closer to the origin first. The moment this extreme is served Situation I occurs.

Situation III $p_t \in I_t$ **and** $0 \in I_t$

IIIx1 $\hat{x}_t \leq \hat{y}_t$ and $t + d(p_t, x_t) \leq L_t^x + x_t$. Go in the direction of the origin (or wait in the origin) until being on the preferred tour. At that moment start to follow the preferred tour first serving x_t.

IIIx2 $\hat{x}_t \leq \hat{y}_t$, $y_t \leq 1.17\hat{x}_t$, and $t + d(p_t, y_t) \leq L_t^y + y_t$. Go in the direction of the origin (or wait in the origin) until being on the anticipating tour. At that moment start to follow the anticipating tour first serving y_t.

IIIx3 $\hat{x}_t \leq \hat{y}_t$. Follow the enforced tour first serving x_t.

IIIy1 $\hat{y}_t \leq \hat{x}_t$ and $t + d(p_t, y_t) \leq L_t^y + y_t$. Go in the direction of the origin (or wait in the origin) until being on the preferred tour. At that moment start to follow the preferred tour first serving y_t.

IIIy2 $\hat{y}_t \leq \hat{x}_t$, $x_t \leq 1.17\hat{y}_t$, and $t + d(p_t, x_t) \leq L_t^x + x_t$. Go in the direction of the origin (or wait in the origin) until being on the anticipating tour. At that moment start to follow the anticipating tour first serving x_t.

IIIy3 $\hat{y}_t \leq \hat{x}_t$. Follow the enforced tour first serving y_t.

Figure 4.2: WA

rightmost unserved extreme y_τ.

At time t cases **Ix1** and **Ix2** can not occur. If at time t case **Iy2** occurs, then $Z^{WA} = t + d(p_t, y_t)$. We use $d(p_t, y_t) < Z^*$ to obtain

$$\frac{Z^{WA}}{Z^*} \leq \frac{t}{Z^*} + \frac{d(p_t, y_t)}{Z^*} < 2.$$

If at time t Situation **II** occurs, then we focus on time t' at which WA serves the extreme closest to the origin, causing Situation **I** to occur. Clearly, WA is ρ-competitive if it can recover a preferred tour at t' (cases **Iy1** and **Ix1**). If the preferred tour can not be recovered, then we distinguish four situations.

- *Case* **Ix2** *occurs at* t' *and* $\hat{y}_t \leq \hat{x}_t$.
  Thus, $Z^{WA} = t + d(p_t, y_t) + d(x_t, y_t)$ and $Z^* \geq \hat{y}_t + d(x_t, y_t)$. We use $d(p_t, y_t) < d(x_t, y_t) < Z^*$ to obtain

$$\frac{Z^{WA}}{Z^*} \leq \frac{t + d(x_t, y_t)}{Z^*} + \frac{d(p_t, y_t)}{Z^*} < 2.$$

- *Case* **Ix2** *occurs at* t' *and* $\hat{x}_t \leq \hat{y}_t$.
  Thus, $Z^{WA} = t + d(p_t, y_t) + d(x_t, y_t)$ and $Z^* \geq \hat{x}_t + d(x_t, y_t)$. We use $d(p_t, y_t) + d(x_t, y_t) < x_t + d(x_t, y_t) \leq Z^*$ to obtain

$$\frac{Z^{WA}}{Z^*} \leq \frac{t}{Z^*} + \frac{d(p_t, y_t) + d(x_t, y_t)}{Z^*} < 2.$$

- *Case* **Iy2** *occurs at* t' *and* $\hat{y}_t \leq \hat{x}_t$.
  Thus, $Z^{WA} = t + d(p_t, x_t) + d(x_t, y_t)$ and $Z^* \geq \hat{y}_t + d(x_t, y_t)$. We use $d(p_t, x_t) < d(x_t, y_t) < Z^*$ to obtain

$$\frac{Z^{WA}}{Z^*} \leq \frac{t + d(x_t, y_t)}{Z^*} + \frac{d(p_t, y_t)}{Z^*} < 2.$$

- *Case* **Iy2** *occurs at* t' *and* $\hat{x}_t \leq \hat{y}_t$.
  For this situation, we have to take the behaviour of the WA-server before time t into account. Since case **Iy2** occurs at t' we have that I_t and x_t are to the right of the origin. So, at time τ Situation **I** or Situation **II** has occurred. In both cases WA starts moving to the left. Thus, $Z^{WA} \leq \tau + d(p_\tau, x_t) + d(x_t, y_t)$ and $Z^* \geq \hat{x}_t + d(x_t, y_t)$. We use $d(p_\tau, x_t) < Z^*$ to obtain

$$\frac{Z^{WA}}{Z^*} \leq \frac{\tau + d(x_t, y_t)}{Z^*} + \frac{d(p_\tau, x_t)}{Z^*} < 2.$$

If at t case **IIIx3** occurs while $p_t \leq 0$, then $Z^{WA} \leq t + 2x_t + y_t$ and $Z^* \geq \hat{x}_t + x_t + y_t$. We have

$$\frac{Z^{WA}}{Z^*} \leq \frac{t}{Z^*} + \frac{2x_t + y_t}{Z^*} \leq 2.$$

If at t case **IIIx3** occurs while $p_t > 0$, then $Z^{WA} = t + 2x_t + y_t + p_t$ and $Z^* \geq \hat{x}_t + x_t + y_t$. We have to take the behaviour of the WA-server before t into account to prove ρ-competitiveness.

If in the time interval $[\tau, t]$ WA does not move to the right, we have that $Z^{WA} = \tau + d(p_\tau, x_t) + x_t + y_t \leq (\tau + x_t + y_t) + (x_t + Y_t) \leq 2Z^*$.

Suppose that WA moves to the right during $[\tau, t]$. We denote the *last* moment in the time interval $[\tau, t]$ at which WA moves to the right by t^r. Obviously, $p_{t^r} > 0$ and $0 \in I_{t^r} = [x_t, y_{t^r}]$. Since WA moves to the right, the last possible cases before time t^r are **IIIx2**, **IIIy1**, and **IIIy3**. We denote y_{t^r} by q.

If the last case before t^r is case **IIIx2** or **IIIy1**, then $Z^{WA} \leq L^y_{t^r} + 2q + 2x_t + y_t = \min\{\rho\hat{q} + (\rho-2)q + (\rho-1)x_t, \rho\hat{x}_t + (\rho-2)x_t + (\rho-3)q\} + 2q + 2x_t + y_t \leq \rho\hat{x}_t + \rho x_t + (\rho-1)q + y_t$. If $q \leq y_t$ we have that $Z^{WA} \leq \rho\hat{x}_t + \rho x_t + \rho y_t \leq \rho Z^*$. Now suppose that $q > y_t$. If in the optimal solution q is served after x_t, then $Z^* \geq \hat{x}_t + x_t + q$. Clearly, $Z^{WA} \leq \rho\hat{x}_t + \rho x_t + \rho q \leq \rho Z^*$.

If $q > y_t$ and the optimal tour serves x_t before y_t and q before x_t, then $Z^* \geq 2q + 2x_t + y_t$. Using $p_t < q$, we have $Z^{WA} = t + 2x_t + y_t + p_t \leq 2Z^*$. In case the optimal tour serves y_t before x_t we have $Z^* \geq t + x_t + y_t$ and therefore $Z^{WA} \leq 2Z^*$.

If the last case before t^r is case **IIIy3**, then by definition $\hat{q} \leq \hat{x}_t$. Clearly, $Z^* \geq \hat{q} + q + x_t$. The last request presented before t^r must be x_t or q. We distinguish four situations.

- The last request presented before t^r is x_t and $p_\tau \geq 0$. We have $Z^{WA} \leq \tau + 2q + 2x_t + y_t = (\tau + x_t + y_t) + (2q + x_t) \leq 2Z^*$.

- The last request presented before t^r is q and $p_{r_q} \geq 0$. We have $Z^{WA} \leq r_q + 2q + 2x_t + y_t = (2x_t + y_t) + (r_q + q + x_t) \leq 2Z^*$.

- The last request presented before t^r is x_t and $p_\tau < 0$. We have $Z^{WA} \leq \tau + p_\tau + 2q + 2x_t + y_t$. If $q > y_t$ then clearly $\tau + p_\tau + 2q - y_t \leq t$, whence $Z^{WA} \leq t + 2x_t + 2y_t$. In case the optimal tour serves y_t before x_t we have $Z^* \geq t + x_t + y_t$ and therefore $Z^{WA} \leq 2Z^*$. In case the optimal tour serves x_t before y_t and q before x_t then $Z^* \geq 2q + 2x_t + y_t > 2x_t + 2y_t$, which together with $Z^* \geq t$ yields $Z^{WA} \geq 2Z^*$.

  If $q \leq y_t$ or if $q > y_t$ and in the optimal solution q is served after x_t, then we have $Z^* \geq \hat{x}_t + x_t + \max[y_t, q]$. We denote max $[y_t, q]$ by y^+ and min $[y_t, q]$ by y^-. If $\tau + p_\tau \leq \rho\hat{x}_t + (\rho-2)x_t + (\rho-3)y^-$ or equivalently $p_\tau \leq \rho\hat{x}_t + (\rho-2)x_t + (\rho-3)y^- - \tau$, then WA is ρ-competitive. We disregard this situation from now on.

  Case **IIIy2** does not occur at time τ. Therefore, if $x_t \leq 1.17\hat{q}$, then $\tau + x_t - p_\tau > \rho\hat{q} + (\rho-2)q + (\rho-2)x_t$. Since we excluded $p_\tau \leq \rho\hat{x}_t + (\rho-2)x_t + (\rho -$

$3)y^- - \tau$, we obtain

$$\begin{aligned} \tau + x_t &> \rho\hat{q} + (\rho-2)q + (\rho-2)x_t + p_\tau \\ &> \rho\hat{q} + (\rho-2)q + (\rho-2)x_t + \rho\hat{x}_t + (\rho-2)x_t + (\rho-3)y^- - \tau \\ &\geq (\rho-1)\hat{x}_t + (2\rho-4)x_t + (3\rho-5)\hat{q} \\ &> (\rho-1)\hat{x}_t + x_t. \end{aligned}$$

This is a contradiction, since $\hat{x}_t \geq \tau$.

If $x_t > 1.17\hat{q}$, then we have to take the behaviour of WA before τ into account. If in the time interval $[r_q, \tau]$ WA does not move to the left, we have that $Z^{WA} \leq r_q + p_{r_q} + 2q + 2x_t + y_t$. Using $p_{r_q} < 0.54 r_q$ (Lemma 4.7) and $x_t > 1.17\hat{q}$, we obtain

$$\begin{aligned} r_q + p_{r_q} + 2q + 2x_t + y_t &< 1.54 r_q + 2q + 2x_t + y_t \\ &\leq 2.48\hat{q} - 0.06x_t + \rho x_t + \rho y^+ \\ &< \rho x_t + \rho x_t + \rho y^+ \\ &\leq \rho\hat{x}_t + \rho x_t + \rho y^+ \\ &\leq \rho Z^*. \end{aligned}$$

Suppose now that WA moves to the left during $[r_q, \tau]$. We denote the *last* moment in the time interval $[r_q, \tau]$ at which WA moves to the left by t_l. Obviously, $p_{t_l} < 0$ and $0 \in I_{t_l} = [x_{t_l}, q]$. Since WA moves to the left, the last possible cases before time t_l are **IIIx1**, **IIIx3**, and **IIIy2**, and $\tau \geq L^x_{tl} + p_\tau = \min\{\rho\hat{x}_{t_l} + (\rho-2)x_{t_l} + (\rho-1)q, \rho\hat{q} + (\rho-2)q + (\rho-3)x_{t_l}\} + p_\tau$. Using $\hat{x}_{t_l} \leq \hat{q}$ or $x_{t_l} \leq 1.17\hat{q}$, we obtain $\tau > q + p_\tau$. We have

$$Z^{WA} \leq \tau + p_\tau + 2q + 2x_t + y_t = (\tau + x_t + y_t) + (q + p_\tau + x_t + q) < 2Z^*.$$

- The last request presented before t^r is q and $p_{r_q} < 0$. Case **IIIy1** does not occur, so $r_q + p_{r_q} > L^q = \min\{\rho\hat{q} + (\rho-2)q + (\rho-1)x_t, \rho\hat{x}_t + (\rho-2)x_t + (\rho-3)q\}$. Since $r_q + p_{r_q} < \rho\hat{q} + (\rho-2)q + (\rho-1)x_t$, we have

$$r_q + p_{r_q} > \rho\hat{x}_t + (\rho-2)x_t + (\rho-3)q, \tag{4.19}$$

implying

$$p_{r_q} > \rho\hat{x}_t + (\rho-2)x_t + (\rho-3)q - r_q. \tag{4.20}$$

Case **IIIy2** neither occurs at time r_q, so $r_q + x_t - p_{r_q} > \rho\hat{q} + (\rho-2)q + (\rho-2)x_t$ or $x_t > 1.17\hat{q}$. Suppose first that $r_q + x_t - p_{r_q} > \rho\hat{q} + (\rho-2)q + (\rho-2)x_t$. Using (4.20), we have

$$\begin{aligned} r_q + x_t &> \rho\hat{q} + (\rho-2)q + (\rho-2)x_t + p_{r_q} \\ &> \rho\hat{q} + (\rho-2)q + (\rho-2)x_t + \rho\hat{x}_t + (\rho-2)x_t + (\rho-3)q - r_q \\ &\geq \rho\hat{x}_t + (2\rho-4)x_t + (3\rho-6)q \\ &> \rho\hat{x}_t. \end{aligned}$$

This is a contradiction, since $\hat{x}_t \geq \hat{q}$.

Now suppose $x_t > 1.17\hat{q}$. Using this in (4.19), we obtain

$$\begin{aligned} r_q + p_{r_q} &> \rho\hat{x}_t + (\rho - 2)x_t + (\rho - 3)q \\ &\geq (2\rho - 2)1.17\hat{q} + (\rho - 3)q \\ &> 1.54\hat{q}. \end{aligned}$$

Again a contradiction, since by Lemma 4.7 $p_{r_q} < 0.54 r_q$.

If at t case **IIIy3** occurs, then $Z^{WA} \leq t + 2x_t + 2y_t$ and $Z^* \geq \hat{y}_t + x_t + y_t$. We have

$$\frac{Z^{WA}}{Z^*} \leq \frac{t + x_t + y_t}{Z^*} + \frac{x_t + y_t}{Z^*} \leq 2.$$

□

4.3 Algorithms for the OLDARP

In this section we present a $\frac{3+\sqrt{5}}{2}$-competitive algorithm for the NOLDARP in general metric spaces. The algorithm is called WI (for Wait or Ignore). WI is described completely by its behaviour at the moment a new request is given. We denote the set of ignored requests at time t by S. We denote the optimal tour over all requests at time t by T_t^* and the optimal tour over all not ignored unserved requests at time t by T_t^{WI}.

Algorithm Wait or Ignore

- If WI can return to the origin before time $(\frac{1+\sqrt{5}}{2})T_t^*$, then he goes back to the origin and empties the set S. He waits until time $(\frac{1+\sqrt{5}}{2})T_t^*$, then he starts to follow the optimal tour T^{WI} over all yet unserved requests.
- Otherwise, WI continues to follow T^{WI}. The new request is added to the set S. When WI is finished, he empties S and computes the optimal tour over all unserved requests that starts in the origin. WI starts to follow this tour, going to the first request of this tour via the shortest path.

Theorem 4.9. *Algorithm Wait or Ignore is $\frac{3+\sqrt{5}}{2}$-competitive for the NOLDARP in general metric spaces.*

Proof. Let time t be the time at which the last request is given. We distinguish two situations.

- *WI can return to the origin before time $(\frac{1+\sqrt{5}}{2})T_t^*$.*
  We have $Z^{WI} \leq (\frac{1+\sqrt{5}}{2})T_t^* + T^{WI}$. Using $T^{WI} \leq T_t^*$, we have that $Z^{WI} \leq (\frac{3+\sqrt{5}}{2})Z^*$.

- *WI cannot return to the origin before time* $(\frac{1+\sqrt{5}}{2})T_t^*$.
  The destination of the last served request in T^{WI} is denoted by x. Let s_q be the source of the first ride from S served in an optimal solution and $T_q(S)$ the shortest tour starting in $s_q \in S$, serving all rides in S. The last time before t the set S was empty we denote by t_l. We have $Z^{WI} \leq t_l + |T^{WI}| + d(x, s_q) + |T_q(S)|$. All rides in S are released after t_l, so $t_q \geq t_l$ and therefore $Z^* \geq t_l + |T_q(S)|$. WI did not leave the origin before time $(\frac{1+\sqrt{5}}{2})T_{t_l}^*$, so $t_l \geq (\frac{1+\sqrt{5}}{2})|T^{WI}|$. We have

$$\frac{Z^{WI}}{Z^*} \leq \frac{t_l + |T_q(S)|}{Z^*} + \frac{d(x, s_q)}{Z^*} + \frac{|T^{WI}|}{Z^*} \leq \frac{3+\sqrt{5}}{2}.$$

□

5

On-line dial-a-ride problems under a restricted information model

The content of this chapter is joint work with X. Lu, W.E. de Paepe, R.A. Sitters, and L. Stougie, and has appeared in [17].

5.1 Introduction

This chapter focuses on on-line single server dial-a-ride problems in general metric spaces in which it is required to return to the origin after having served all the requests (HOLDARP). For a formal problem definition see Section 2.1.

On-line dial-a-ride problems have been studied in literature before [2], [6]. In these papers it is assumed that the rides are specified completely upon presentation, i.e., both the source and the destination of the ride become known at the same time. We diverge from this setting here, by assuming that at the release of a ride only information about the source is given. At visiting the source, the information about the destination is made available to the servers. For many practical situations our model is closer to reality. Think for example of the problem to schedule an elevator. Here, a ride is the transportation of a person from one floor (the source) to another (the destination), and the release time of the ride is the moment the button on the wall outside the elevator is pressed. The destination of such a ride is revealed only at the moment the person enters the elevator and presses the button inside the elevator. Other examples of such situations in practice are taxi, minibus, and courier services.

We feel that lack of information is often a *choice*, rather than inherent to the problem: additional information *can* be obtained, but this requires investments in information systems. In this chapter we give mathematical evidence that for the problem under study it pays to invest.

We study the on-line single server dial-a-ride problem in which only the source of a ride is presented at the release time of the ride. The destination of a ride is revealed at visiting its source. We call this model the *incomplete ride information model* and refer to the model used in [2] and [6] as the *complete ride information model*. In contrast to the previous chapters we allow the server to have capacity greater than one.

We distinguish two versions of the on-line dial-a-ride problem under the incomplete ride information model. In the first version the server is allowed to preempt any ride at any point, and resume the ride later. In particular the server is allowed to visit the source of a ride and learn its destination without executing the ride immediately. This version we call the *preemptive version*. In the second version, the *non-preemptive* version, a ride has to be executed as soon as the ride has been picked up in the source. In this version we do allow the server to pass a source without starting the ride, in which case he does not learn the destination of the ride at passing the source. We study each version of the problem under various *capacities* of the server. The capacity of a server is the number of rides the server can execute simultaneously. We consider unit capacity, constant capacity $c \geq 2$, and infinite capacity for the server.

In [2] (see Section 2.3.3) a best possible 2-competitive deterministic algorithm is given for the on-line dial-a-ride problem under the complete ride information model, independent of the capacity of the server. In this paper preemption of rides is not allowed. However, the lower bound of 2 comes from a sequence of rides with zero length, an instance of the on-line travelling salesman problem [3](see Section 2.3.1), hence the bound also holds for the problem with preemption. We show that under the incomplete ride information model, no deterministic algorithm can have a competitive ratio smaller than 3, even if preemption is allowed, and independent of the capacity of the server. For the preemptive version, we design an algorithm with competitive ratio matching the lower bound of 3, independent of the capacity of the server. These results are presented in Section 5.2.

If preemption is not allowed, we derive a lower bound of $\max\{c, 1 + \frac{3}{2}\sqrt{2}\}$ on the competitive ratio of any deterministic algorithm, where c is a given fixed capacity of the server. We present a $(2c + 2)$-competitive algorithm for the non-preemptive version. These results are presented in Section 5.3.

We notice that there is no difference between the preemptive version and the non-preemptive version of the problem if the server has infinite capacity, hence we inherit the matching lower and upper bound of 3 of the preemptive version for this case. An overview of the results obtained in this chapter is given in Table 5.1.

The results in this chapter, combined with those from [2], show the effect of having complete knowledge about rides on worst-case performance for on-line dial-a-ride problems. This is an important issue, since in practice complete information is

Table 5.1: Overview of lower bounds (LB) and upper bounds (UB) on the competitive ratio of deterministic algorithms for on-line dial-a-ride problems.

	capacity	LB	UB
complete ride information			
preemption	$1, c, \infty$	2 [3]	2 [2]
no preemption	$1, c, \infty$	2 [3]	2 [2]
incomplete ride information			
preemption	$1, c, \infty$	3	3
no preemption	1	$1 + \frac{3}{2}\sqrt{2}$	4
	c	$\max\{1 + \frac{3}{2}\sqrt{2}, c\}$	$2c + 2$
	∞	3	3

often lacking. Investments in information systems can help to obtain more complete information, and mathematical support is essential in justifying such investments.

We conclude this introduction by referring back to the elevator scheduling problem. We have seen that the typical elevator with only a request button at the wall outside the elevator fits our incomplete ride information model. In an alternative construction of an elevator, the destination buttons could be built outside the elevator, fitting the complete ride information model. Notice that to minimize the latest completion time is not the most natural objective for an elevator.

5.2 The preemptive version

We describe our algorithm SNIFFER, which preempts rides only immediately at the source, just to learn the destinations of the rides: it "sniffs" the rides. Upon visiting the source of a ride for the second time, the ride is completed right away. The algorithm is an adaption of the 2-competitive algorithm for the on-line travelling salesman problem (OLTSP), described in [3]. The proof of 3-competitiveness of SNIFFER borrows parts of the proof in the latter paper. The algorithm is described completely by the actions it takes at any moment t at which the server either arrives in the origin or receives a new request. When SNIFFER computes an optimal tour over a set of requests or rides, we always assume that this tour includes the origin. We use $|T|$ to denote the length of a tour T.

Algorithm SNIFFER

(1) **The server is in the origin at t.**
If the set S of yet unvisited sources is non-empty, compute the optimal travelling salesman tour $T_{\text{TSP}}(S)$ on the points in S, and start following $T_{\text{TSP}}(S)$. Just learn the destinations of the rides with sources in S, without starting to execute any of these rides.
If $S = \emptyset$ and the set R of rides yet to be executed is non-empty, compute the optimal dial-a-ride tour $T_{\text{DAR}}(R)$ on the rides in R. Also compute the optimal dial-a-ride tour $T_{\text{DAR}}(\sigma_{\leq t})$ on all rides requested in $\sigma_{\leq t}$. If $t = 2|T_{\text{DAR}}(\sigma_{\leq t})|$,

start following $T_{\text{DAR}}(R)$. If $t < 2|T_{\text{DAR}}(\sigma_{\leq t})|$, remain idle. If no new requests arrive before time $2|T_{\text{DAR}}(\sigma_{\leq t})|$ start following $T_{\text{DAR}}(R)$ at time $2|T_{\text{DAR}}(\sigma_{\leq t})|$.

(2) **The server is on a tour $T_{\text{TSP}}(S)$ at t when a new ride is released.** Let p_t denote the location of the server at time t. If the new ride, say $\sigma_k = (t, s_k, ?)$ (the question mark indicating the unknown destination), is such that $d(s_k, O) > d(p_t, O)$, then return to the origin via the shortest path, ignoring all rides released while travelling to the origin.
If $d(s_k, O) \leq d(p_t, O)$, ignore the new ride until the origin is reached again and proceed on $T_{\text{TSP}}(S)$.

(3) **The server is on a tour $T_{\text{DAR}}(R)$ at t when a new ride is released.** Return to the origin as soon as possible via the shortest path, and ignore rides released in the mean time. If the server is executing a ride, the ride is finished before returning to the origin.

Theorem 5.1. *Algorithm* SNIFFER *is* 3*-competitive for the preemptive OLDARP problem under the incomplete ride information model, independent of the capacity of the server.*

Proof. Let $T_{\text{DAR}}(\sigma)$ be the optimal tour over all rides of σ. It is sufficient to prove that for any sequence σ the server can always be in the origin at time $2|T_{\text{DAR}}(\sigma)|$ to start the final tour on the yet unserved rides. He will then always finish this tour before time $3|T_{\text{DAR}}(\sigma)| \leq 3\text{OPT}(\sigma)$. This is obviously true for any sequence σ consisting of only one ride. We assume it holds for any sequence of $m-1$ rides, and prove that then it also holds for any sequence σ of m rides. Let $\sigma_m = (t_m, s_m, d_m)$ be the last ride in σ (notice that the destination d_m is not given to the on-line algorithm until the moment the source s_m is visited).

(1) Suppose the server is in O at t_m, and $S \neq \emptyset$. He starts tour $T_{\text{TSP}}(S)$ and returns to O at time $t_m + |T_{\text{TSP}}(S)| \leq 2|T_{\text{DAR}}(\sigma)|$.

(2) Suppose the server is in p_{t_m} following $T_{\text{TSP}}(S)$. If $d(O, s_m) \leq d(O, p_{t_m})$, σ_m is added to a set Q of rides ignored since the last time the server was in O. Let $s_q \in Q$ be the source of a ride visited first in an optimal solution. Since this ride was ignored, the server was at least $d(O, s_q)$ away from the origin at time t_q, and hence had moved at least this distance on tour $T_{\text{TSP}}(S)$. Thus, the server returns in O before $t_q + |T_{\text{TSP}}(S)| - d(O, s_q)$. Back in O the server commences on $T_{\text{TSP}}(Q)$. Let $P_q(Q)$ be the path of minimum length that starts in s_q, ends in O, and visits all sources in Q. Obviously, $|T_{\text{TSP}}(Q)| \leq d(O, s_q) + |P_q(Q)|$. Hence the server is back in the origin after visiting all sources no later than $t_q + |T_{\text{TSP}}(S)| - d(O, s_q) + d(O, s_q) + |P_q(Q)| = t_q + |T_{\text{TSP}}(S)| + |P_q(Q)| \leq 2|T_{\text{DAR}}(\sigma)|$, since, clearly, $|T_{\text{DAR}}(\sigma)| \geq t_q + |P_q(Q)|$, and $|T_{\text{DAR}}(\sigma)| \geq |T_{\text{TSP}}(S)|$.

If $d(O, s_m) > d(O, p_{t_m})$, the server returns to O immediately, arriving there at $t_m + d(O, p_{t_m}) < t_m + d(O, s_m) \leq |T_{\text{DAR}}(\sigma)|$. Back in O the server computes and starts following an optimal TSP tour over the yet unvisited sources, which has a length of at most $|T_{\text{DAR}}(\sigma)|$. Hence the server is back in O again before time $2|T_{\text{DAR}}(\sigma)|$.

If the server was already moving towards the origin because a ride was released before σ_m that was further away from the origin than the on-line server, then the arguments above remain valid.

(3) Suppose the server is on a tour $T_{\text{DAR}}(R)$ at time t_m, or moving towards O because of another ride released before t_m. Let $t(R)$ be the time at which the server started $T_{\text{DAR}}(R)$. Then $R \subset \sigma_{\leq t(R)}$ and $t(R) = 2|T_{\text{DAR}}(\sigma_{\leq t(R)})|$, by induction. Thus, the server is back in O before time $3|T_{\text{DAR}}(\sigma_{\leq t(R)})|$. There, it starts a tour $T_{\text{TSP}}(S)$ over a set S of unvisited sources, being again back in O before time $3|T_{\text{DAR}}(\sigma_{t(R)})| + |T_{\text{TSP}}(S)| = \frac{3}{2}t(R) + |T_{\text{TSP}}(S)|$. We need to show that this is not greater than $2|T_{\text{DAR}}(\sigma)|$.

Let s_q be the first ride from S served in an optimal solution and $P_q(S)$ the shortest path starting in $s_q \in S$, ending in O, and visiting all sources in S. Clearly, $|T_{\text{DAR}}(\sigma)| \geq t_q + |P_q(S)|$ and $|T_{\text{TSP}}(S)| \leq 2|P_q(S)|$. Since all rides in S are released after $t(R)$, $t_q \geq t(R)$. Therefore, $|T_{\text{DAR}}(\sigma)| \geq t(R) + |P_q(S)|$ and

$$\frac{3}{2}t(R) + |T_{\text{TSP}}(S)| \leq 2t(R) + 2|P_q(S)| \leq 2|T_{\text{DAR}}(\sigma)|.$$

□

We show that SNIFFER is a best possible deterministic algorithm for the preemptive version of the OLDARP problem, even if SNIFFER uses preemption only at the source of rides.

Theorem 5.2. *No deterministic algorithm can have a competitive ratio smaller than $3 - \epsilon$ for the OLDARP problem under the incomplete ride information model, independent of the capacity of the server, where ϵ is arbitrarily small.*

PROOF. For the proof of this theorem we use a commonly applied setting of a two-person game, with an adversary providing a sequence of rides, and an on-line algorithm serving the rides (see [5]). Typically, the outcome of the algorithm is compared by the solution value the adversary achieves himself on the sequence, which usually is the optimal off-line solution value. We consider the OLDARP problem under the incomplete ride information model where the on-line server has infinite capacity. Let ALG be a deterministic on-line algorithm for this problem. We will construct an adversarial sequence σ of requests for rides. We restrict the adversary by giving his server capacity 1. We will prove that ALG can not be better than $3-\epsilon$-competitive for this restricted adversary model, where ϵ is arbitrary small.

The metric space $M = (X, d)$ is a graph with vertex set $X = \{x_1, x_2, \ldots, x_{n^2}\} \cup O$ and the distance function d, where $d(O, x_i) = 1$ and $d(x_i, x_j) = 2$ for all $x_i, x_j \in X \setminus O$. To facilitate the exposition we denote point x_i by i.

At time 0 there is one ride in each of the n^2 points in $X \setminus \{O\}$. If the on-line server visits the source i of a ride at time t with $t \leq 2n^2 - 1$, then the destination turns out to be i as well, and at time $t+1$, a new ride with source i is released.

In this way, the situation remains basically the same for the on-line server until time $2n^2$. We may assume that at some moment t^*, with $2n^2 - 1 < t^* \leq 2n^2$, there is exactly one ride $\sigma_i = (t_i, i, d_i)$ in each of the points i. Without loss of generality we assume that the vertices i are labelled in such a way that $t_1 \leq \cdots \leq t_{n^2}$.

Thus, at time t^* the on-line server still has to complete exactly n^2 rides. We partition the set of n^2 vertices into n sets: $I_k = \{(n-1)k+1, \ldots, nk\}$, $k = 1, \ldots, n$. Within each of these sets we order the vertices by the on-line server's first visit to them after time t^*. Let b_{kj}, $j \in \{1, \ldots, n\}$ be the jth vertex in this order in I_k. Now we define for all $k \in \{1, \ldots, n\}$ the destination of the ride in vertex b_{kj} as b_{k1} for $j = 1$ and $b_{k,j-1}$ for all $j \in \{2, \ldots, n\}$. Notice that the destination of ride σ_i only depends on the tour followed by the on-line server until he picks up the ride to look at its destination. For the on-line server this means that n of the n^2 rides can be served immediately since the source equals the destination. For the other $n^2 - n$ rides the server finds out that the destination of the rides he just picked up is another point that he already visited after time t^*. Therefore, $n^2 - n$ points will have to be visited by the on-line server at least twice after time t^*. Hence, the completion time for the on-line server is at least $t^* + 4(n^2 - n) - 1 + 2n > 6n^2 - 2n - 2$.

We will now describe the tour made by the adversary. Given our definition of t^* we have that $t_{n^2} \leq t^* \leq 2n^2$. Since the on-line server needs at least 2 time units to move from a point i to another point i', it follows that $t_i \leq 2i$, for all $i \in \{1, \ldots, n^2\}$. The adversary waits until time $2n$ and then starts to serve the rides $\sigma_1, \ldots, \sigma_n$, by visiting the sources in reversed order of $b_{11}, \ldots, b_{1n}$. The rides with equal source and destination are served immediately at arrival in the point. This takes the adversary $2n$ time units. At time $4n$ the adversary starts serving the rides $\sigma_{n+1}, \ldots, \sigma_{2n}$, and then at time $6n$ the rides $\sigma_{2n+1}, \ldots, \sigma_{3n}$, etc. Continuing like this the server has completed all the rides and is back in the origin at time $2n^2 + 2n$.

Hence, the competitive ratio is bounded from below by $(6n^2 - 2n - 2)/(2n^2 + 2n)$, which can be made arbitrarily close to 3 by choosing n large enough. □

5.3 The non-preemptive version

For the non-preemptive version we design an algorithm, called BOUNCER, because the server always "bounces" back to the source, once a ride is completed. This algorithm uses as a subroutine the 2-competitive algorithm for the OLTSP problem from [3].

Algorithm BOUNCER

Perform the OLTSP algorithm on the sources of the rides. This algorithm outputs a tour T. The BOUNCER server follows tour T, until a source is visited. There he executes the ride, and returns to the source via the shortest path. As soon as the server arrives in the source again, he continues to follow T.

Theorem 5.3. *Algorithm* BOUNCER *is* $(2c + 2)$*-competitive for the OLDARP* *problem under the incomplete ride information model, where c is the capacity of the server.*

PROOF. Consider any request sequence σ. Since the OLTSP algorithm is 2-competitive, and in any solution for the OLDARP problem all sources have to be visited, $\mathrm{OPT}(\sigma) \geq |T|/2$. Let $D = \sum_{i:\sigma_i \in \sigma} d(s_i, d_i)$, then also $\mathrm{OPT}(\sigma) \geq D/c$. The completion time of the BOUNCER server is at most $T + 2D \leq 2\mathrm{OPT}(\sigma) + 2c\mathrm{OPT}(\sigma)$. □

Corollary 5.4. *Algorithm* BOUNCER *is* 4*-competitive for the OLDARP problem under the incomplete ride information model, if the capacity of the server is* 1*.*

Theorem 5.5. *No non-preemptive deterministic on-line algorithm can have a competitive ratio smaller than $c - \epsilon$ for the OLDARP problem under the incomplete ride information model, where the capacity of the server c is a constant, and where $\epsilon > 0$ is arbitrarily small.*

PROOF. Consider an instance of the OLDARP problem on a star graph with $K >> c$ leaves, where the origin is located in the center of the star, and each leaf has distance 1 to the origin. At time 0, cK rides are released, all with their source in the origin, and each of the leaves being destination of c rides. Thus, there are K sets of c identical rides each, hence the instance has an optimal solution value of $2K$.

Any on-line server can carry at most c rides at a time. The instance is constructed in such a way that, until time $2(cK - c^2)$, any time the on-line server is in the origin he has all different rides (rides with different destinations). It is clear that this can indeed be arranged, given that the on-line server can not distinguish between the rides until he picks them up at the source and he has no possibility to preempt, not even in the source. At time $2(cK - c^2)$ the on-line server can have served at most $cK - c^2$ rides, and hence at least c^2 rides remain yet to be served, requiring an extra of at least $2c$ time units. Hence the completion time of any on-line server is at least $2(cK - c^2) + 2c$.

Therefore, the competitive ratio is bounded from below by

$$\frac{2(cK - c^2) + 2c}{2K} = \frac{cK - c^2 + c}{K}.$$

For any $\epsilon > 0$ we can choose K large enough for the theorem to hold. □

Together with Theorem 5.1 this theorem shows that for servers with capacity greater than 3, the best possible deterministic on-line algorithm for the non-preemptive version of the problem has a strictly higher competitive ratio than SNIFFER for the preemptive problem, in which the server can have any capacity. The following theorem shows that this phenomenon also occurs for lower capacities of the server.

Theorem 5.6. *No non-preemptive deterministic algorithm can have a competitive ratio smaller than* $1+\frac{3}{2}\sqrt{2}-\epsilon \approx 3.12$ *for the OLDARP problem under the incomplete ride information model, where* ϵ *is arbitrarily small.*

PROOF. First we consider the OLDARP problem under the incomplete ride information model when the on-line server has capacity 1. Then we will sketch how to extend the proof for any capacity c.

Let ALG be a non-preemptive deterministic on-line algorithm for this problem. The metric space is a star graph with $2n$ leaves. All leaves have length 1. The center of the star is the origin O and the leaf vertices are denoted by a_i $(i = 1 \dots n)$, and b_i $(i = 1 \dots n)$. On every leaf a_i and b_i there is an additional vertex a'_i or b'_i at distance $1 - \alpha$ $(0 < \alpha < 1)$ from the origin, where α is a fixed number that we choose appropriately later.

We give the following sequence σ of rides. At time zero there are three rides in each point a_i and b_i, $i = 1 \dots n$. If the on-line server visits a source, then the destination turns out to be the same as the source. This kind of rides are called empty rides. One time unit later the ride is replaced by a new ride with the same source. Every source that is visited by the on-line server before time $4n$ is handled in this way. Sources visited after this time are not replaced.

Let t_i^a (resp. t_i^b) be the last moment before time $4n - 4$ that the on-line server is in point a_i (resp. b_i). We set $t_i^a = 0$ (resp. $t_i^b = 0$) if a_i (resp. b_i) is not visited before $4n - 4$. Without loss of generality we assume that $t_i^a \le t_j^a$, $t_i^b \le t_j^b$, $t_i^a \le t_j^b$, and $t_i^b \le t_j^a$, for all $1 \le i < j \le n$. Any server needs at least 2 time units to travel from one leaf to another, implying that $t_i^a \le 4(i-1)$ and $t_i^b \le 4(i-1)$, for all $i \in \{1, \dots, n\}$.

We refer to the three rides that were released latest in a leaf as the *decisive rides* and define them as follows. In each point b_i two of the decisive rides are empty and one has destination a_i. In point a_i one of the decisive rides is empty, one has destination a'_i, and one is either empty or has destination O. With these rides, the on-line server is unable to distinguish between the points a_i and b_i, and since we did not distinguish between these points before, we may assume that, after time $4n-4$, the on-line server visits point a_i before point b_i. The first ride that the server picks up in point a_i is the ride to a'_i. The first ride the server picks up in point b_i is the ride to a_i. We distinguish between two cases. In the first case the on-line server executes the ride from b_i to a_i before it picks up the second ride in a_i. In this case the second ride is a ride to the origin. Otherwise, the second ride being picked up in a_i is empty.

In the first case the on-line server needs at least 10 time units (from origin to origin) to serve all the rides connected to each pair a_i, b_i, whereas in the optimal off-line solution only $4 + 2\alpha$ time units are required. In the second case the on-line server needs at least $8 + 2\alpha$ time units, whereas in the optimal off-line solution 4 time units are required. The optimal off-line strategy starts a tour at time 0, first serving a_1 and b_1, then serving a_2 and b_2, etc. Empty rides are served without taking any extra time.

The on-line server cannot start with the decisive rides until time $4n-4$. Assume that he is in the origin at time $4n-5$ and then moves to point a_1. Now we consider the contribution of a pair a_i, b_i in the total time needed for the on-line and the off-line server and we take the ratio of the two. For fixed α ($0 < \alpha < 1$) this ratio becomes at least

$$\min\left\{\tfrac{10+(4n-5)/n}{4+2\alpha}, \tfrac{8+2\alpha+(4n-5)/n}{4}\right\} \quad > \quad \min\left\{\tfrac{14}{4+2\alpha}, \tfrac{12+2\alpha}{4}\right\} - \tfrac{2}{n}.$$

Optimizing over α yields $\alpha = -4+3\sqrt{2}$. Hence, for the competitive ratio we find

$$\frac{\mathrm{ALG}(\sigma)}{\mathrm{OPT}(\sigma)} > 1 + \frac{3}{2}\sqrt{2} - \frac{2}{n}.$$

For any ϵ we can choose n large enough for the theorem to hold in the case of a unit capacity server. If the capacity of the server is c, with $c \geq 1$, we just give c copies of the same sequence σ simultaneously. An on-line server cannot benefit from this extra capacity in combining rides from different pairs a_i, b_i. The on-line server will have to do the rides in a specific point in the same order as before. For example the first c rides that the on-line server picks up in a_i are rides to a'_i. Hence, the completion time for the on-line server cannot be smaller than in the capacity 1 case, and the off-line server can complete in exactly the same time. □

Corollary 5.7. *No non-preemptive deterministic on-line algorithm can have a competitive ratio smaller than* $\max\{1 + \frac{3}{2}\sqrt{2}, c\} - \epsilon$ *for the OLDARP problem under the incomplete ride information model, where the capacity of the server c is a constant and* $\epsilon > 0$ *is arbitrarily small.*

5.4 Discussion

In [2] and [6] the competitive ratio measures the cost of having no information about the release times of future rides. In this discussion we show how we can measure the cost of having no information about the destinations of the rides through the competitive ratio.

Suppose that at time 0 the release times and the location of the sources of the rides are given, but the information about the destinations is again revealed only at visiting the sources.

Both SNIFFER and BOUNCER use the on-line algorithm of Aussiello et al. [3] for a TSP tour along the sources. In case all sources of the rides and the release times are known, an optimal TSP tour over the sources, that satisfies the release time constraints, can be computed (disregarding complexity issues). In this way SNIFFER and BOUNCER gain an additive factor of 1 on their competitive ratio, making SNIFFER 2-competitive and BOUNCER $2c+1$-competitive.

Notice that the lower bound of $c - \epsilon$ on the competitive ratio for the non-preemptive problem in Theorem 5.5 is obtained through a sequence of rides all with release time 0, thus this lower bound is completely due to the lack of information about the destinations of the rides.

The rides in the sequence giving the lower bound of $1 + \frac{3}{2}\sqrt{2}$ for the non-preemptive problem in Theorem 5.6 have release times no larger than $4n-5$. Taking the unserved rides at time $4n-5$ as an instance given at time 0, shows that the competitive ratio is at least $\min\{\frac{10}{4+2\alpha}, \frac{8+2\alpha}{4}\}$. Optimizing over α yields a lower bound of $\frac{1}{2} + \frac{1}{2}\sqrt{11} \approx 2,15$. Thus due to the lack of information about destinations only, any algorithm will not be able to attain a ratio of less than $\max\{\frac{1}{2}+\frac{1}{2}\sqrt{11}, c-\epsilon\}$.

In the lower bound construction for the preemptive problem in Theorem 5.2 the adversary stops giving requests at time $2n^2$. Take the set of rides unserved by any on-line algorithm at that time as an instance with release time 0. Following the proof of Theorem 5.2 any on-line algorithm will need $4n^2 - 2n$, whereas an optimal tour takes $2n^2$, yielding a lower bound of 2.

Notice that the above lower bounds are established on sequences where all rides have release time 0. For the preemptive version of the problem this is clearly sufficient, since the performance of SNIFFER matches the lower bound. However, for the non-preemptive version higher lower bounds might be obtained using diverse release times of rides.

Bibliography

[1] F. Afrati, S. Cosmadakis, C.H. Papadimitriou, G. Papageorgiou, and N. Papakonstantinou, *The complexity of the traveling repairman problem*, Theoretical Informatics and Applications **20** (1986), 79–87.

[2] N. Ascheuer, S.O. Krumke, and J. Rambau, *Online dial-a-ride problems: Minimizing the completion time*, Proceedings of the 17th International Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Computer Science, vol. 1770, Springer, 2000, pp. 639–650.

[3] G. Ausiello, E. Feuerstein, S. Leonardi, L. Stougie, and M. Talamo, *Algorithms for the on-line travelling salesman*, Algorithmica **29** (2001), 560–581.

[4] M. Blom, S.O. Krumke, W.E. de Paepe, and L. Stougie, *The online-TSP against fair adversaries*, INFORMS Journal on Computing **13** (2001), 138–148.

[5] A. Borodin and R. El-Yaniv, *Online computation and competitive analysis*, Cambridge University Press, Cambridge, 1998.

[6] E. Feuerstein and L. Stougie, *On-line single-server dial-a-ride problems*, Theoretical Computer Science **268** (2001), 91–105.

[7] A. Fiat and G.J. Woeginger (eds.), *Online algorithms: The state of the art*, Lecture Notes in Computer Science, vol. 1442, Springer, 1998.

[8] M.R. Garey and D.S. Johnson, *Computers and intractability: A guide to the theory of NP-completeness*, W.H. Freeman Company, San Francisco, 1979.

[9] R.L. Graham, E.L. Lawler, J.K. Lenstra, and A.H.G. Rinnooy Kan, *Optimization and approximation in deterministic sequencing and scheduling: a survey*, Annals of Discrete Mathematics **5** (1979), 287–326.

[10] G. Gutin and A.P. Punnen (eds.), *The traveling salesman problem and its variations*, Kluwer, Dordrecht, 2002.

[11] R.M. Karp, *Reducibility among combinatorial problems*, Complexity of Computer Computations (R.E. Miller and J.W. Thatcher, eds.), Plenum Press, New York, 1972, pp. 85–103.

[12] E. Koutsoupias and C. Papadimitriou, *Beyond competitive analysis*, Proceedings of the 35th Annual IEEE Symposium on the Foundations of Computer Science, 1994, pp. 394–400.

[13] S.O. Krumke, *Online optimization: Competitive analysis and beyond*, Habilitationsschrift, Technische Universität Berlin, 2001.

[14] S.O. Krumke, W.E. de Paepe, D. Poensgen, and L. Stougie, *News from the online traveling repairman*, Proceedings of the 26th International Symposium on Mathematical Foundations of Computer Science, Lecture Notes in Computer Science, vol. 2136, Springer, 2001, pp. 487–499.

[15] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys (eds.), *The traveling salesman problem: a guided tour of combinatorial optimization*, Wiley, Chichester, 1985.

[16] M. Lipmann, L. Laura, A. Marchetti-Spaccamela, S.O. Krumke, W.E. de Paepe, D. Poensgen, and L. Stougie, *Non-abusiveness helps: An o(1)-competitive algorithm for minimizing the maximum flow time in the online traveling salesman problem*, Proceedings of the 5th International Workshop on Approximation Algorithms for Combinatorial Optimization, Lecture Notes in Computer Science, vol. 2462, Springer, 2002, pp. 200–214.

[17] M. Lipmann, X. Lu, W.E. de Paepe, R. Sitters, and L. Stougie, *Online dial-a-ride problems under a restricted information model*, Proceedings of the 10th Annual European Symposium on Algorithms, Lecture Notes in Computer Science, vol. 2461, Springer, 2002, pp. 674–685.

[18] W.E. de Paepe, *Computer-aided complexity classification of dial-a-ride problems*, Master's thesis, University of Amsterdam, Faculty of Economics and Econometrics, 1998.

[19] ______, *Complexity results and competitive analysis for vehicle routing problems*, Ph.D. thesis, Eindhoven University of Technology, 2002.

[20] W.E. de Paepe, J.K. Lenstra, J. Sgall, R.A. Sitters, and L. Stougie, *Computer-aided complexity classification of dial-a-ride problems*, INFORMS Journal on Computing, to appear.

[21] R.A. Sitters, *The minimum latency problem is* NP*-hard for weighted trees*, Proceedings of the 9th Mathematical Programming Society Conference on Integer Programming and Combinatorial Optimization, Lecture Notes in Computer Science, vol. 2337, Springer, 2002, pp. 230–239.

[22] D.D. Sleator and R.E. Tarjan, *Amortized efficiency of list update and paging rules*, Communications of the ACM **28** (1985), 202–208.

[23] J.N. Tsitsiklis, *Special cases of travelling salesman and repairman problems with time windows*, Networks **22** (1992), 263–282.

Samenvatting

In dit proefschrift behandelen we routeringsproblemen in een dynamische omgeving. Routeringsproblemen treden op wanneer een route gezocht moet worden die een gegeven aantal plaatsen bezoekt. Denk bijvoorbeeld aan een handelsreiziger die een aantal klanten moet bezoeken. Om tijd en kosten te besparen probeert hij de kortste route te vinden waarmee hij alle klanten aandoet. Dit probleem staat bekend als het *handelsreizigersprobleem*. Een ander voorbeeld is een koerier die pakketten moet ophalen en vervolgens ergens anders moet bezorgen. Ook hij wil zo snel mogelijk klaar zijn en probeert de kortste route te bepalen. Dit probleem staat bekend als een *dial-a-ride* probleem.

Routeringsproblemen komen zowel in een dynamische omgeving als in een statische omgeving voor. We noemen een omgeving statisch als alle opdrachten van het routeringsprobleem bekend zijn op het moment dat een route bepaald moet worden. Denk bijvoorbeeld aan een koerier die aan het begin van zijn werkdag een lijst heeft waarop al zijn opdrachten voor die dag staan. Hij kan zijn route voor die dag in één keer bepalen.

In een dynamische omgeving worden de opdrachten van het routeringsprobleem in de loop van de tijd gegeven. Er is niets bekend over eventueel toekomstige opdrachten en op elk willekeurig moment zijn alleen de opdrachten die in het verleden zijn gegeven bekend. Terwijl aan een oplossing wordt gewerkt en er een bepaalde route wordt gevolgd kunnen er nieuwe opdrachten binnenkomen. Er moet dan een nieuwe route bepaald worden. Denk bijvoorbeeld aan een koerier met een mobiele telefoon. Terwijl hij de tot dan toe bekende opdrachten uitvoert kan hij gebeld worden voor nieuwe opdrachten. We noemen een routeringsprobleem in een dynamische omgeving een on-line routeringsprobleem en een routeringsprobleem in een statische omgeving een off-line routeringsprobleem.

In dit proefschrift bestuderen we on-line routeringsproblemen waarin we zoeken naar de route die de tijd waarop alle opdrachten zijn uitgevoerd minimaliseert. We onderscheiden hierin twee gevallen. In het eerste geval moet de route eindigen in het startpunt, in het tweede geval is het eindpunt van de route vrij. Ook bestuderen we routeringsproblemen waarin we zoeken naar de route die de som van de completeringstijden van alle opdrachten minimaliseert. De completeringstijd van een opdracht is het vroegste moment dat een opdracht geheel is uitgevoerd.

Voor deze problemen beschouwen we on-line algoritmen (oplossingsmethoden) die op elk moment en in alle mogelijke situaties de te volgen route bepalen. De kwaliteit van een on-line algoritme bepalen we aan de hand van competitiviteitsanalyse. Daarbij worden de kosten van de route van een on-line algoritme vergeleken met de kosten van de optimale route. Hierbij onderscheiden we de vrije optimale route en de optimale route die aan bepaalde 'eerlijkheidscriteria' moet voldoen. Een algoritme heeft een competitiviteitsratio ρ als het voor iedere willekeurige instantie een route geeft waarvan de kosten niet meer zijn dan ρ maal de kosten van de optimale route. We proberen enerzijds algoritmen te vinden met een zo laag mogelijke competitiviteitsratio, aan de andere kant proberen we ondergrenzen te vinden op competitiviteitsratio's. Een ondergrens op de competitiviteitsratio voor een bepaald probleem houdt in dat geen enkel on-line algoritme een lagere competitiviteitsratio kan hebben voor dat probleem. Een ondergrens geeft een indicatie van de kosten van het niet kennen van eventuele toekomstige opdrachten ongeacht het gebruikte on-line algoritme.

In dit proefschrift beschouwen we routeringsproblemen in verschillende metrische ruimtes. We onderscheiden de halflijn, de lijn en algemene metrische ruimtes. We onderscheiden twee soorten algoritmen. Het eerste soort algoritme is het 'ijverige' algoritme: tijdens de te volgen route mag er niet gedraald of gewacht worden zolang er niet-uitgevoerde opdrachten zijn. Het tweede soort algoritme is het 'normale' algoritme: er mag gedraald of gewacht worden tijdens de te volgen route. We geven een overzicht van de best bekende ondergrenzen en de best bekende algoritmen uit dit proefschrift of uit de literatuur voor bovengenoemde problemen.

Curriculum vitae

Maarten Lipmann was born on December 13, 1968, in Lisse, the Netherlands. In 1986, he received his Gymnasium diploma from the Stedelijk Gymnasium in Haarlem. In 1987, Maarten graduated at the Loreauville Highschool, Louisiana, the United States. He came back to the Netherlands and started to study Physics at Delft University of Technology in 1987. A year later Maarten switched to the University of Amsterdam, where he studied Operations Research and Management. In December 1999 he received his Master's degree. In May 2001 he started as a Ph.D. student at the department of Mathematics and Computer Science of Eindhoven University of Technology under the supervision of dr. L. Stougie and prof.dr. J.K. Lenstra. The results of his research are presented in this thesis.